User's Guide

FileFixer™ for V8

Fix your MicroStation® V8 files automatically!

Version 8.14

Axiom
1805 Drew Street
Clearwater, Florida 33765
1-727-442-7774 voice
1-727-442-8344 fax
Info@AxiomInt.com
www.AxiomInt.com

Copyright © 1985-2014 Axiom.
All rights reserved.
This document was last modified on 4 August 2014 10:07 AM.

Limited Warranty — (Excerpted from the "Axiom End User Software License Agreement", see the full text of your "Axiom End User Software License Agreement" for complete details.) LICENSEE is advised to test the SOFTWARE thoroughly before relying on it. The SOFTWARE is provided "as is" without warranty of any kind, either expressed or implied, including, but not limited to the implied warranties of merchantability and fitness for a particular purpose and the entire risk as to the quality and performance of the SOFTWARE is with LICENSEE.

Governing Law — This Agreement ("Axiom End User Software License Agreement") shall be governed by the laws of the State of Florida, United States of America. Venue for the resolution of any dispute arising out of this Agreement shall be Pinellas County, Florida, USA or, if that is not possible, the nearest court of competent jurisdiction thereto.

FileFixer™ and *Problem Element Viewer*™ are trademarks of Axiom International.
Windows® is a registered trademark of Microsoft Corporation.
Intergraph® is a registered trademark of Intergraph Corporation.
MicroStation® and MDL® are registered trademarks of Bentley Systems, Incorporated.
Intergraph Raster File Formats copyright © 1994 Intergraph Corporation.
Used with permission.
ZLIB copyright © 1995-1998 Jean-loup Gailly and Mark Adler.
Used with permission.

Table of Contents

Chapter 1 — Introduction ---------- 13
The Tradition Continues ---------- 13
Features and Benefits ---------- 14
Urgent note ---------- 14

Chapter 2 — Installation ---------- 15
Before you install ---------- 15
Typical installation from CD ---------- 15
Congratulations ---------- 21

Chapter 3 — Uninstalling ---------- 22
Uninstalling Axiom products ---------- 22

Chapter 4 — Quick Start ---------- 23
Quick Start Objectives ---------- 23
Prerequisites ---------- 23
Demonstration Version ---------- 23
Quick Start Step 1 — The Basics ---------- 23
When should I use *FileFixer?* ---------- 23
Factory Defaults ---------- 24
Quick Start Step 2 — Automatic Design File Repair ---------- 25
Quick Start Step 3 — *FileFixer*'s Report ---------- 27
Quick Start Step 4 — Searching for Problems ---------- 29
Quick Start Step 5 — What's Next? ---------- 31
Problem Element Viewer ---------- 31

Chapter 5 — Running *FileFixer* ---------- 32
Getting started with *FileFixer* ---------- 32
The main dialog box ---------- 32
Mode ---------- 32
Which Files? ---------- 33
Report File ---------- 34
<Start> ---------- 34
<Display Report> ---------- 34
The File menu ---------- 35
File | Save Edited File List as List File… ---------- 35
File History ---------- 36
File | Exit ---------- 36
The Settings menu ---------- 36
Change Settings ---------- 36
Load Settings... ---------- 36
Restore Factory Defaults... ---------- 37
Save Settings ---------- 37

Save Settings As... 37
Current Settings File: 37
FIXER_INI (custom location for your settings file) 37
Help | Contents... 38
Help | About... 38

Chapter 6 — The Change Settings Menu 39
Change Settings Menu 39
General... 39
Search... 39
Repair... 40
Report File... 40

Chapter 7 — General Settings 41
The General Settings Dialog Box 41
Alternate temporary files directory 41
Skip checking the file if the first element is not a valid element. 42
Initialize “Which Files” field on startup with: 42
When repairing a design file or cell library 42
Backup/repaired file name 43
Use the original file’s extension for the repaired file. 44
Add on the repair file extension instead of replacing it. 44
Set modification date and time of fixed file to that of the original file. 45
Search for Problems Mode 45
Skip checking the file if there is a seal-of-approval. 45
Write seal-of-approval information to the design file. 45

Chapter 8 — Search Settings 46
Factory defaults 46
Cell 46
Cell is nested within itself. 46
Non-shared cell class map is incorrect. 47
Null name in cell library. 47
Shared cell definition with no shared cell instance. 47
Shared cell instance has no shared cell definition. 47
Complex 48
Complex bit is on for a non-component element. 48
Complex bit is off for a component element. 48
Complex element contains disallowed element type. 49
Component on different level than its header. 49
Component graphic group doesn’t match header. 49
Gap exists in 2D complex shape or chain. 49
Shared cell instance nested in a shared cell definition with the same cell name. 50
Design File 50
Severity of errors to be reported: 50
Build a list containing the names of the design files with ... 51
Dimension 51
Dimension element with no data points or associative points. 51
Element 52
Element class is illegal (must be less than 7). 52

Element's and model's dimensions are different. ---- 52
Graphical element in non-graphical data. ---- 52
Graphical element is invisible. ---- 53
H-bit is set when it shouldn't be. ---- 53
Illegal element type. ---- 53
Reserved bits are set in element header. ---- 53
Preserve Distributed Graphics System (DGS) deleted elements. ---- 53
Unused bit is set in the "properties" word. ---- 54
Element is beyond the edge of the design plane. ---- 54
Enter-Data-Fields ---- 54
Text element has more EDFs than characters. ---- 55
Headers ---- 55
Design file header is on the wrong level. ---- 55
Missing vital element (type 9). ---- 55
Invalid text style values in the design file header. ---- 55
Levels ---- 56
Level is hidden. ---- 56
Multiple master file level tables. ---- 56
Multiple level numbers. ---- 57
Unused reference file level table. ---- 57
Linkage ---- 57
"Application required for element modification" flag in a TriForma linkage. ---- 57
Attribute offset error. ---- 58
Duplicate text formatting linkage in a text node. ---- 58
MicroStation application-specific attribute linkage found. ---- 58
New line flag is missing from a text element linkage in a text node. ---- 58
Shared cell instance dependency linkage is pointing to a shared cell definition with a different cell name. ---- 59
Shared cell instance dependency link to its shared cell definition is broken. ---- 59
Siamese attribute linkage (linkage overlaying the next element). ---- 59
User linkage is not a multiple of 4 words. ---- 60
Models ---- 60
Model is locked. ---- 60
Range ---- 61
Complex header range doesn't match components' range. ---- 61
Element size, excluding lines and line strings, is zero in all dimensions. ---- 61
Low range greater than high range. ---- 62
Z range is not zero in a 2D element ---- 62
Range of graphic element is wrong. ---- 62
Reference File ---- 63
Duplicate reference file slot encountered ---- 63
___ levels of reference files should be processed. ---- 64
Missing reference file. ---- 64
Reference file attachment using slot zero. ---- 64
Reference file is hidden. ---- 64
Structured Storage ---- 65
Missing or damaged model header or index data. ---- 65
Damaged graphical data. ---- 66
Damaged non-graphical data. ---- 66
Damaged non-model data. ---- 66
Damaged Project Explorer data. ---- 66
Missing graphical data. ---- 67

Missing non-graphical data. ---- 67
Missing non-model data. ---- 67
Text ---- 68
Text element doesn't contain any visible characters. ---- 68
Text element uses non-existent font. ---- 68
Text string overflows the element's length. ---- 68
Vertices ---- 69
Duplicate b-spline poles detected in a closed b-spline. ---- 69
Line string's vertex is off the design plane. ---- 69
Too few vertices on line string, shape, curve, etc. ---- 70
Too many vertices on line string, shape, curve, etc. ---- 70

Chapter 9 — Repair Settings ---- 71
Factory defaults ---- 71
B-spline ---- 71
Open the closed b-spline that has duplicate b-spline poles ---- 71
Cell ---- 72
Adjust non-shared cell class map. ---- 72
Correct null cell names (type 2s only) by setting the H-bit. ---- 72
Delete cell nested within itself. ---- 73
Delete shared cell definition with no shared cell instances. ---- 73
Handle missing shared cell definitions. ---- 73
Complex ---- 74
Add line to bridge gap in 2D complex shape or chain. ---- 74
Correct complex bit errors. ---- 74
Delete shared cell instance nested in a shared cell definition with the same cell name. ---- 75
Move components to same level as complex header. ---- 75
Set graphic group of component to match its header. ---- 75
Handle illegal components in complex element. ---- 75
Dimension ---- 76
Remove dimension element with no data points or associative points. ---- 76
Element ---- 77
Delete graphical element in non-graphical data. ---- 77
Fix invisible graphical element by: ---- 77
Fix reserved bits by clearing them? ---- 78
Remove elements with too few vertices. ---- 78
Remove illegal element types. ---- 78
Set the element's dimension to match the model's dimension. ---- 78
Set illegal class to: ---- 79
Turn off illegal H-bits. ---- 79
Delete problem elements based on the error they generate. ---- 79
ElementSize ---- 80
Filter factor ---- 80
Fix element size warnings. ---- 80
Enter-Data-Fields ---- 81
Correct EDF/text length mismatches. ---- 81
Headers ---- 82
Correct the design file header's level ---- 82
Replace Non-Model Header or Table ---- 82
Levels ---- 84

Delete duplicate master file level numbers by: ---- 84
The error number is 429.Delete unused master file level tables ---- 84
Delete unused reference file level tables. ---- 85
Display hidden levels ---- 85
Repair elements on corrupted levels ---- 85
Linkage ---- 87
Add new line flag missing in a text element linkage that is part of a text node. ---- 87
Delete duplicate text formatting linkage in a text node. ---- 88
Remove siamese linkage. ---- 88
Clear “application required for element modification” flag in TriForma linkage. -- 88
Models ---- 89
Unlock models. ---- 89
Range ---- 90
Repair range of complex header elements. ---- 90
Repair range of displayable elements. ---- 90
Reference File ---- 91
Delete missing reference file attachment. ---- 91
Display hidden reference files. ---- 91
Set duplicate slot number to an unused slot number ---- 91
Structured Storage ---- 92
Internal File Structure Damage ---- 92
Internal Compression Damage ---- 93
Internal Data Damage ---- 94
Text ---- 98
Correct the number of characters field in text element ---- 98
Vertices ---- 99
Fix line string’s vertex which is off the edge of the design plane by ---- 99
Set number of vertices to maximum allowed ---- 99

Chapter 10 — Report File Settings ---- 100
Factory defaults ---- 100
Report file settings, factory defaults ---- 101
Format: ---- 102
Extension ---- 102
Append the extension ---- 102
Report file style ---- 103
Put the report summary ---- 104
What to include ---- 104
One or Multiple Report Files ---- 105
Program to display and print the ASCII report file: ---- 108

Chapter 11 — MicroStation Application Settings ---- 109
MicroStation Applications Settings dialog box... ---- 109
TriForma settings ---- 109

Chapter 12 — Reports ---- 111
Report Summary ---- 111
The File Position Number ---- 112
The Unique Element ID ---- 112
The *FileFixer* Error Message and Error Number ---- 113

Chapter 13 — Error Messages ---- 114
Operational errors and messages ---- 114
Suspicious Conditions ---- 116
V8-specific Informational Messages (Errors 201–267) ---- 120
Structured Storage ---- 120
Minor Errors ---- 123
V8-specific Warnings ---- 129
Future Problems — File is in Danger ---- 132
Serious V8-specific Errors ---- 134
Serious Errors — Problems are Likely ---- 136
Fatal V8-specific Errors ---- 143
Fatal Errors ---- 146

Chapter 14 — Batch Mode ---- 148
What is batch mode? ---- 148
Why use batch mode? ---- 148
How do you run *FileFixer* in batch mode? ---- 148
Valid options ---- 149
Examples ---- 150

Chapter 15 — Preventive Maintenance ---- 151
Nightly *FileFixer* maintenance ---- 151
Setting up a typical nightly *FileFixer* maintenance run ---- 151
Scheduling ---- 151
Create the settings file for nightly *FileFixer* run ---- 152
Set up the Windows scheduler ---- 153
Set up the *FileFixer* MSBATCH command ---- 155

Chapter 16 — Tips ---- 156
About yesterday's backup... ---- 156
Using *FileFixer* with Design File History ---- 156
V7 files ---- 156
If *FileFixer* fails to handle an error ---- 157
Controlling the types of errors which are reported ---- 157
Our commitment ---- 157

Chapter 17 — Choose Files to Process ---- 158
Using Axiom's "Choose Files to Process" box ---- 158
Adding files to the list ---- 159
File and Edit menus ---- 159
Browse ProjectWise ---- 159
Browse for files ---- 161
Search for files ---- 162
Search in: ---- 162
Include subfolders. ---- 162
Search for files using: ---- 162
Search for files by date: ---- 163

{Search Now} ---163
{Add Files} ---163
{Close} ---163
Search for Files: Advanced tab ---163
Drag-and-drop files from Windows Explorer ---164
Dragging folders (not files) into the list ---164
Remove selected files ---164
Remove all files ---164
Remove Missing Files ---164
Save list to file ---165
Files to Process ---165
Right-clicking on the list ---165
Selecting files in the list ---165
Removing files from the list ---166
Version ---166
{OK} ---166
{Cancel} ---166
Advanced Commands ---166

Chapter 18 — Glossary ---168
Specialized terms used in this guide ---168

Chapter 19 — We'll Make (Almost) Any Enhancements You Want ---175
We welcome your suggestions ---175

Chapter 20 — New Features and Fixes ---176
8.14a — 6 August 2014 ---176
New! Replace a deleted or badly damaged design file header element. ---176
New! Fix shared cells with bad dependency linkages. ---176
Improved: Replace a missing shared cell definition from an available cell library. 176
New! Automatically fix elements with too many vertices. ---176
Additional Improvements ---176
8.13a — 12 February 2013 ---177
New! *FileFixer* fully tested and certified with MicroStation V8i SELECTseries 3. 177
New! Automatically fix missing or damaged model headers. ---177
New! Fix reference file attachments that use slot zero. ---177
New! Fix damaged Project Explorer data. ---177
New! Remove TriForma modification lock from elements. ---177
Improved: Find and fix gaps in 2D complex shapes and chains. ---178
Additional improvements ---178
8.12d — 4 May 2012 ---178
8.12c — 26 January 2012 ---178
8.12b — 24 June 2011 ---178
8.12a — 1 October 2010 ---179
Replace a non-model table ---179
Delete graphical data in the non-graphical data area ---179
Additional enhancements ---179

8.11a — 8 January 2010----------179
Corrupt linkage enhancements ----------179
8.10a — 4 December 2009----------180
ProjectWise integration----------180
Additional enhancements ----------180
8.9a — 18 June 2009 ----------180
Duplicate reference file slot number----------180
Other enhancements in this release ----------180
8.8a — 26 December 2008----------181
8.7i — 2 May 2008 ----------181
Replace the Design File Header Element ----------181
Element has Different Dimension than the Model's Dimension----------182
Unused Reference File Level Table----------182
Duplicate master file level numbers ----------182
Other Enhancements in this Release----------182
8.7h — 30 November 2007 ----------183
Newly Supported V8-specific Fatal Error ----------183
Other Enhancements in this Release----------184
8.7g — 31 August 2007 ----------185
Multiple Level Tables (Duplicate Level Definitions) ----------185
Locked Models ----------186
Hidden Levels ----------187
Hidden Reference File Attachments ----------188
Additional Enhancements ----------188
8.7f — 31 May 2007----------189
8.7e — 9 March 2007----------190
Help at the touch of a button----------190
Customer-driven enhancements ----------191
Text Style corruption----------192
8.7d — 24 November 2006 ----------193
V8 elements are compressed.----------194
V8 design file organization is new. ----------194
Structured Storage directory corruption----------194
New Search and Repair breakthroughs----------195
8.7c — 28 August 2006 ----------195
8.7b — 8 June 2006----------197
8.7a — 31 March 2006 ----------198
8.6b — 6 December 2005----------200
8.6a — 18 August 2005 ----------201
169 — Non-zero Z range found in a 2D element. ----------201
8.5b — 24 February 2005----------201
8.5a — 10 September 2004 ----------202
8.4b — 7 May 2004----------203
8.4a — 10 April 2004----------203
8.3a — 30 January 2004----------204
Search and Repair enhancements ----------205
8.2a — 20 December 2003----------206
Search and Repair enhancements ----------206
Additional enhancements ----------207
8.1a — 25 July 2003----------208
More TriForma-specific defaults----------208
Additional enhancements in *FileFixer for V8* version 8.1a----------209

8.0e — 23 May 2003 ----------209
8.0d — 17 December 2002 ----------210
New Settings File Format: Plain Text ----------211
Report Enhancements----------211
Severity of Errors to be Reported ----------211
TriForma Enhancements ----------212
Miscellaneous Enhancements ----------212
8.0c — 7 November 2002----------212

Chapter 1 — Introduction

The Tradition Continues

Since 1987 Axiom has provided the Intergraph and (later) the MicroStation communities with independent and objective file analysis and repair. By popular demand, this tradition continues with *FileFixer for V8*.

It is not Axiom's position that *any* version of MicroStation introduces erroneous data into design files. MicroStation/J is quite robust. Nevertheless, our V7 clients still regularly report design file problems to us and use *FileFixer* regularly. How can this be?

Even if MicroStation (V7 or V8) never introduced any erroneous data into design files, factors beyond Bentley's control can still impact your files: acts of God, acts of Bill Gates, your operating system, your network software, your network hardware, your server, your CPU, your hard drives and your electric power company.

And sure enough, the tradition continues with MicroStation V8.

FileFixer for V7 is the best guarantee that your V7 files will import into V8 healthy and whole.

Once you begin production with MicroStation V8, you enter the realm of *FileFixer for V8*.

Like its V7 counterpart, *FileFixer for V8* will evolve in response to client requests. The "New Features and Fixes" section of this guide chronicles the evolution of *FileFixer for V8*.

We look forward to hearing from you about *FileFixer for V8*.

To get started immediately, see the Quick Start.

Features and Benefits

- Fixes V8 design files and cell libraries at the touch of a button
- Fixes 2D and 3D MicroStation files in batch
- Fixes internal structured storage problems
- Processes models automatically
- Identifies and repairs structured storage (compound file) problems (new for V8)
- Generates complete reports of the anomalies found and the actions done to fix them
- Includes the most-used features from the V7 version of *FileFixer*

Note: See the "New Features and Fixes" section of this guide for a complete description of the latest *FileFixer* enhancements.

Urgent note

It is important to create a backup copy of a design file before repairing it with *FileFixer*. There are two reasons for this.

1. If anything non-optimal is later discovered about the fixed file, you will still have the original file to work with. Although *FileFixer* generally does a good job of repairing design files, the user is still the final judge as to whether the "fixed" file is truly superior to the original file.
2. If you need assistance with design file repair, Axiom Support personnel will need a copy of the original unfixed design file.

Chapter 2 — Installation

Before you install

This chapter will walk you through a typical Axiom product installation. These instructions apply to both MicroStation V7 and V8. V7 products are used in the examples.

To install one or more Axiom products, you need:

- The latest Axiom product CD; and
- A license file on floppy disk for the product(s) you are about to install.

Please shut down all MicroStation sessions on your computer before continuing.

Typical installation from CD

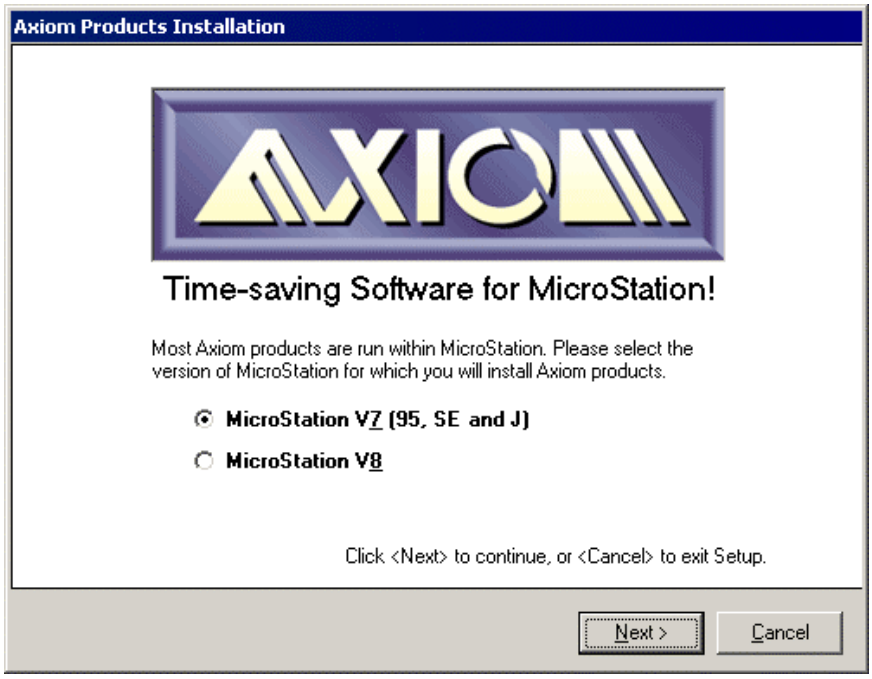
1. Insert the Axiom product CD into your CD drive.
2. The dialog box below should come up automatically. This may take a few moments depending on computer and CD drive speed.

CONDITIONAL: In some cases Windows® is set to suppress auto-running executables. If the installation process does not begin automatically when you insert the CD, you will need to start the process manually. From the Windows Start Menu, choose “Run…” and type in the following:

```
D:\Setup.exe
```

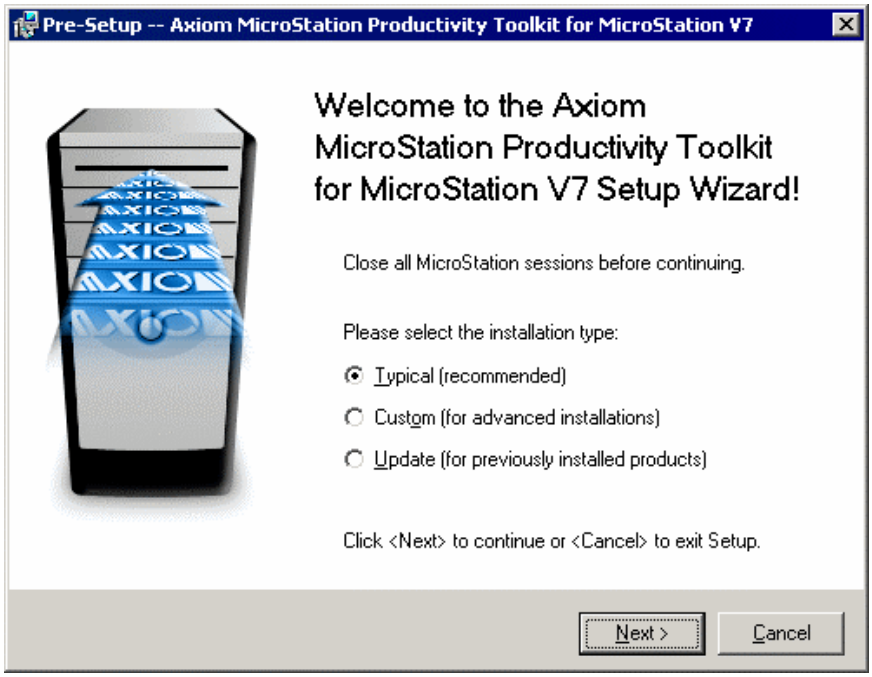
where “D” is your CD drive letter.

3. Click the {OK} button and the first installation dialog box (below) will appear.


Axiom Products Installation
Time-saving Software for MicroStation!
Most Axiom products are run within MicroStation. Please select the version of MicroStation for which you will install Axiom products.
MicroStation V7 (95, SE and J)
MicroStation V8
Click <Next> to continue, or <Cancel> to exit Setup.
Next >
Cancel

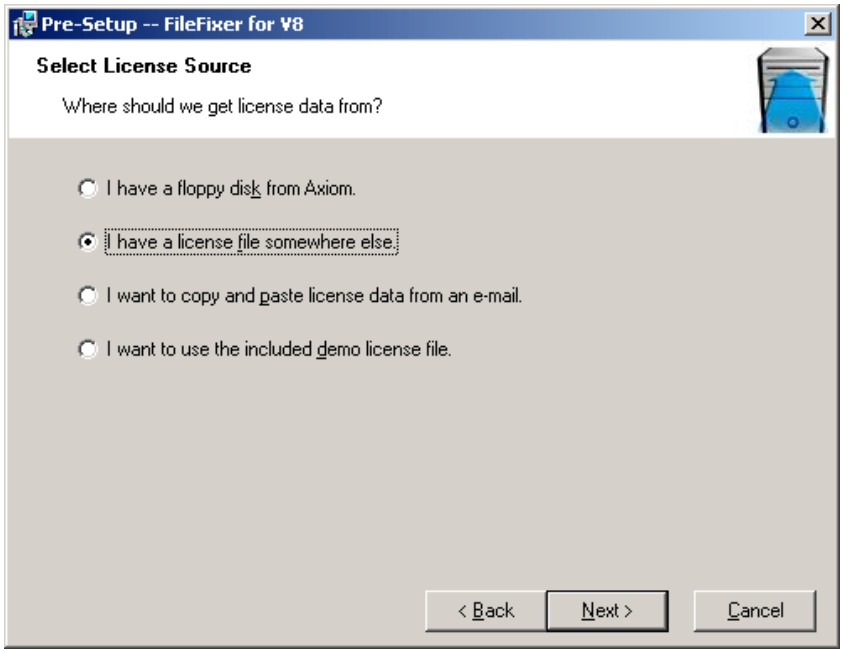
4. Select MicroStation V7 or MicroStation V8 and click {Next}.

5. Choose which type of installation you want to perform. The “Typical” installation is best for most situations and asks the user to make very few decisions.


Pre-Setup -- Axiom MicroStation Productivity Toolkit for MicroStation V7
Welcome to the Axiom MicroStation Productivity Toolkit for MicroStation V7 Setup Wizard!
Close all MicroStation sessions before continuing.
Please select the installation type:
Typical (recommended)
Custom (for advanced installations)
Update (for previously installed products)
Click <Next> to continue or <Cancel> to exit Setup.
Next >
Cancel

6. These instructions will describe only the “Typical” installation path. Choose “Typical” and click {Next}.

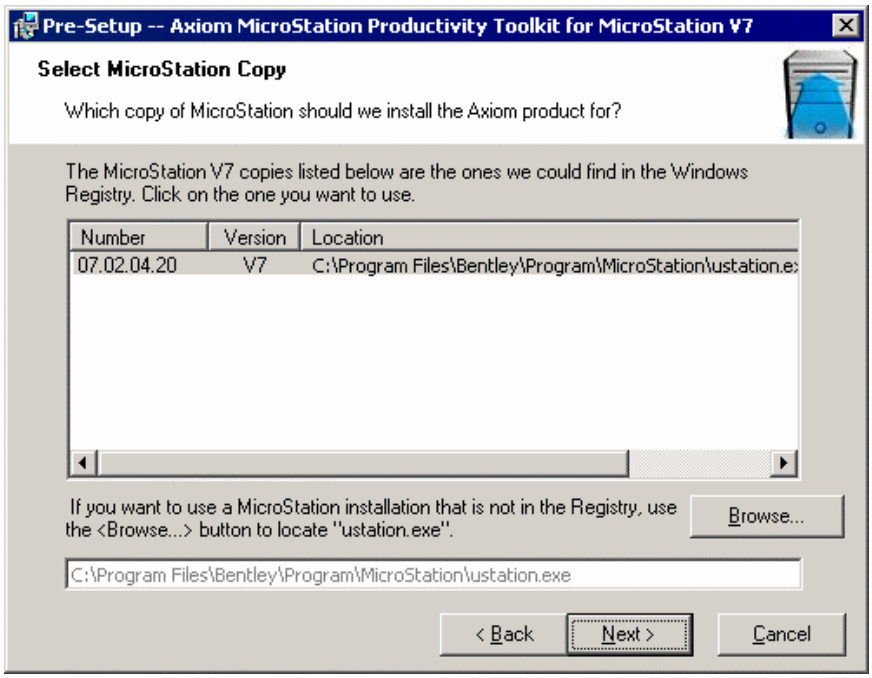
7. Choose the license file for this installation. Toggle on “I have a license file somewhere else.” And press {Next}.


Pre-Setup -- FileFixer for V8
Select License Source
Where should we get license data from?
I have a floppy disk from Axiom.
I have a license file somewhere else.
I want to copy and paste license data from an e-mail.
I want to use the included demo license file.
< Back
Next >
Cancel

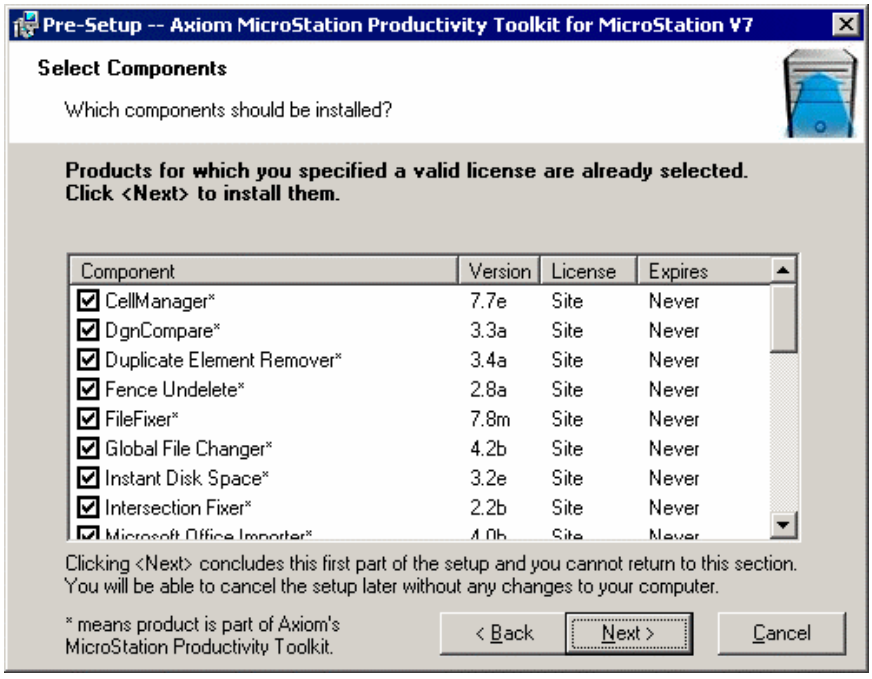
8. Browse to the “axiom.lic” file on the CD and click {Open}. The licenses in the specified license file will display in the next dialog box. Press {Next}.

   Although you have already selected the MicroStation family (V7 or V8) for which you want to install Axiom tools, it is common for one computer to have more than one instance of MicroStation installed on it. For example, you might have two versions of MicroStation/J installed (07.01.04.16 and 07.02.04.20). You must choose which of the two you want to associate your Axiom tools with.

9. Highlight the version of MicroStation you want to associate the Axiom tools with and click {Next}.


Pre-Setup -- Axiom MicroStation Productivity Toolkit for MicroStation V7
Select MicroStation Copy
Which copy of MicroStation should we install the Axiom product for?
The MicroStation V7 copies listed below are the ones we could find in the Windows Registry. Click on the one you want to use.
Number
Version
Location
07.02.04.20
V7
C:\Program Files\Bentley\Program\MicroStation\ustation.e
If you want to use a MicroStation installation that is not in the Registry, use the <Browse...> button to locate "ustation.exe".
Browse...
C:\Program Files\Bentley\Program\MicroStation\ustation.exe
< Back
Next >
Cancel

Since this is a “Typical” installation, all programs for which an Axiom license was found will already be “checked-on” to be installed. Other items that are also automatically “checked-on” for installation are the product User’s Guides, individual product sample files and Axiom’s MicroStation pull-down menu.



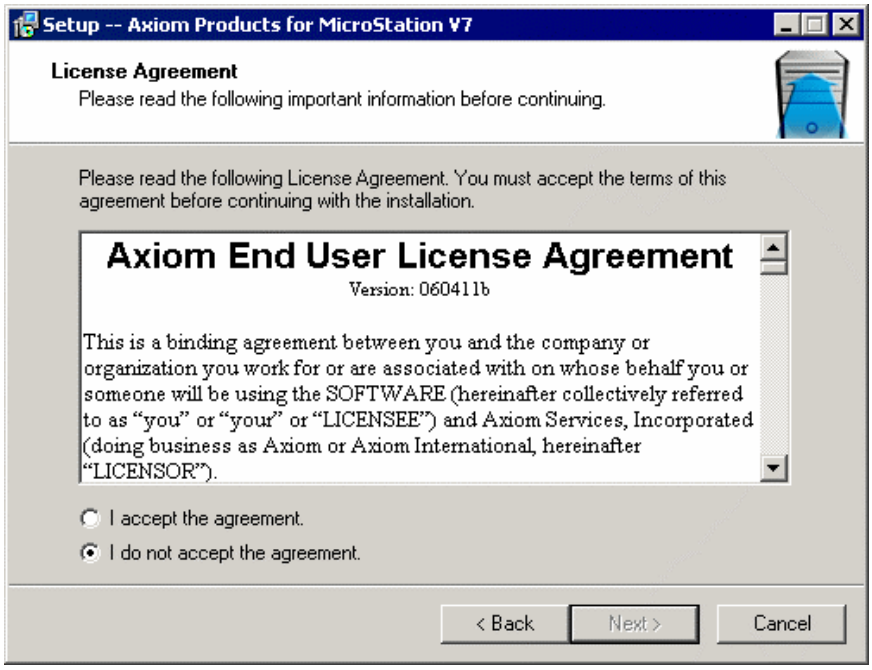
10. To proceed with default selections click {Next}.

    You've now completed the Pre-Setup.

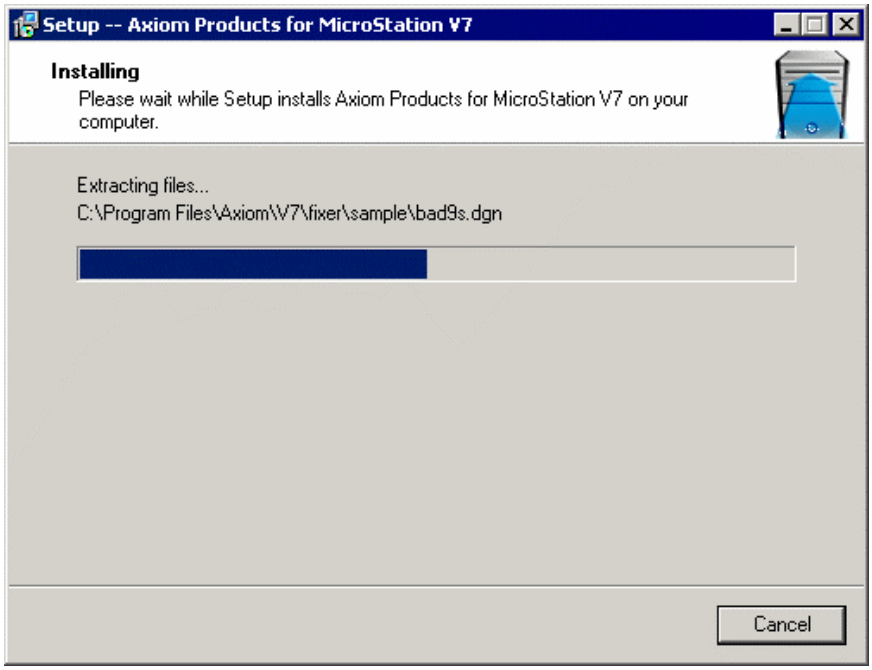
Setup -- Axiom Products for MicroStation V7
Continuing Setup of Axiom Products for MicroStation V7
Click <Next> to continue, or <Cancel> to exit Setup.
Next >
Cancel

11. Click {Next} to proceed with the installation.

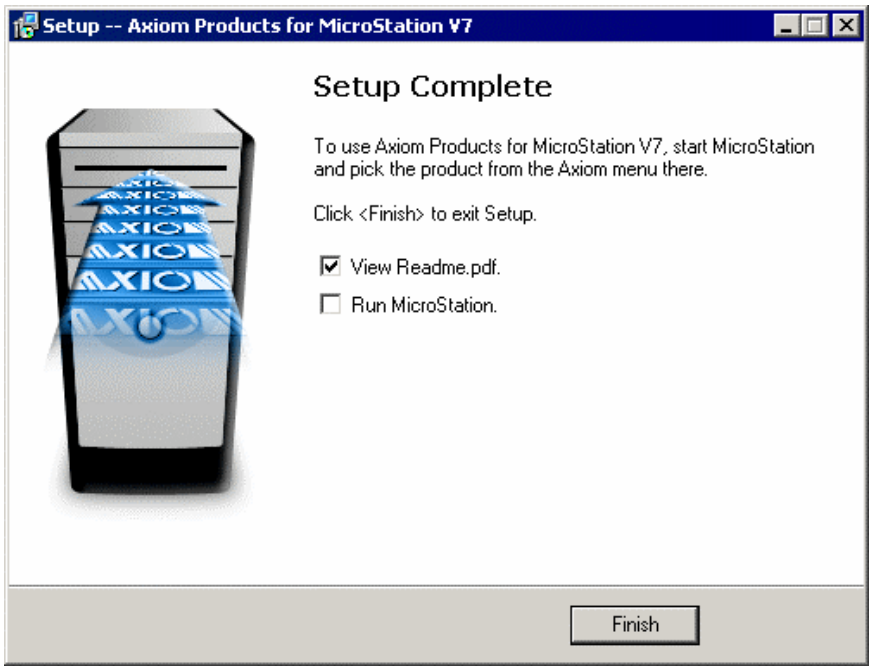
12. The {Next} button is grayed-out unless you accept the agreement.


Setup -- Axiom Products for MicroStation V7
License Agreement
Please read the following important information before continuing.
Please read the following License Agreement. You must accept the terms of this agreement before continuing with the installation.
Axiom End User License Agreement
Version: 060411b
This is a binding agreement between you and the company or organization you work for or are associated with on whose behalf you or someone will be using the SOFTWARE (hereinafter collectively referred to as "you" or "your" or "LICENSEE") and Axiom Services, Incorporated (doing business as Axiom or Axiom International, hereinafter "LICENSOR").
I accept the agreement.
I do not accept the agreement.
< Back
Next >
Cancel

If you accept the license and click {Next}, the installation will begin. You can click {Cancel} at any time to abort the installation.


Setup -- Axiom Products for MicroStation V7
Installing
Please wait while Setup installs Axiom Products for MicroStation V7 on your computer.
Extracting files...
C:\Program Files\Axiom\V7\fixer\sample\bad9s.dgn
Cancel

13. This is the final dialog box. It shows that the installation is complete. By default the “View Readme.pdf” option is toggled on and will display the Readme.pdf document when you click {Finish}.


Setup -- Axiom Products for MicroStation V7
Setup Complete
To use Axiom Products for MicroStation V7, start MicroStation and pick the product from the Axiom menu there.
Click <Finish> to exit Setup.
View Readme.pdf.
Run MicroStation.
Finish

Congratulations

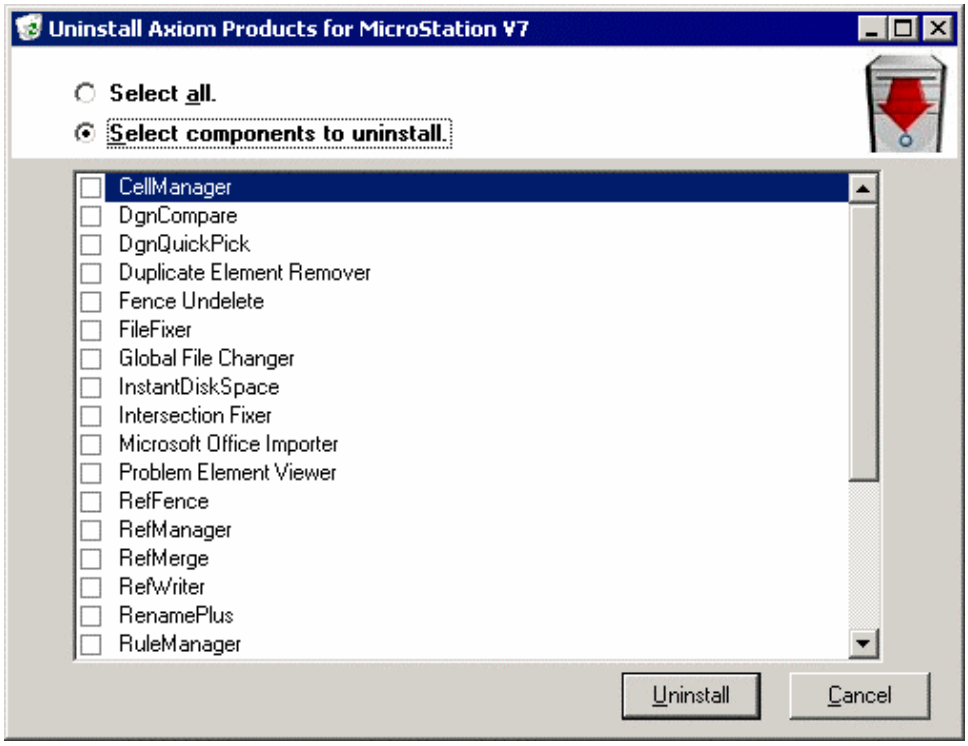
That’s it! You have successfully completed the installation of your Axiom tools.

Chapter 3 — Uninstalling

Uninstalling Axiom products

Axiom products can be uninstalled easily through the Windows® Start Menu.

1. Go to {Start}, *Programs* | *Axiom*.
2. Select “Uninstall V7 products” or “Uninstall V8 products” depending on the products you want to uninstall.



3. Choose “Select all.” or “Select components to uninstall.” If you choose “Select components to uninstall.” then each product in the list can be manually checked or unchecked. This allows for any combination of products to be uninstalled. Only installed products will be shown in the list.

Note: At the bottom of the list of products you also have the option to uninstall sample design files, user’s guides and even the Axiom pull-down menu.

4. Once you’ve made your selections, click the [Uninstall] button to complete the uninstall process.

Chapter 4 — Quick Start

Quick Start Objectives

The purpose of this Quick Start is to introduce you to the basic features of *FileFixer* so that you can immediately use the program.

This Quick Start should take less than 30 minutes.

Prerequisites

1. MicroStation V8 version 8.1.2.15 or later.
2. *FileFixer* must be installed on your computer. You can verify this by selecting the "Axiom" menu from MicroStation's main menu bar, then select "FileFixer". If "FileFixer" does not appear in the "Axiom" menu, follow the instructions in the *Installation* chapter of this User Guide.

Note: You must have "write" permission so that you can modify your practice files.

Demonstration Version

When processing files which are greater than 100KB with a demonstration version of *FileFixer,* only about half of the problems found will be reported in detail, although the Report Summary will indicate the actual quantity of each type of problem found during processing.

Licensed versions of *FileFixer* do not have these restrictions.

Quick Start Step 1 — The Basics

When should I use *FileFixer?*

FileFixer is most commonly used to:

- Automatically repair problem files.
- Search for problems as preventative maintenance (to avoid future problems) or to confirm files are "healthy" and problem-free.

FileFixer has two modes of operation to accommodate these common uses:

When you select "Automatic Design File Repair" mode, *FileFixer* searches for problems *and* automatically fixes them.

When you select "Search for Problems" mode, *FileFixer* searches for problems and reports what is found.

This Quick Start will introduce you to these common uses of *FileFixer*.

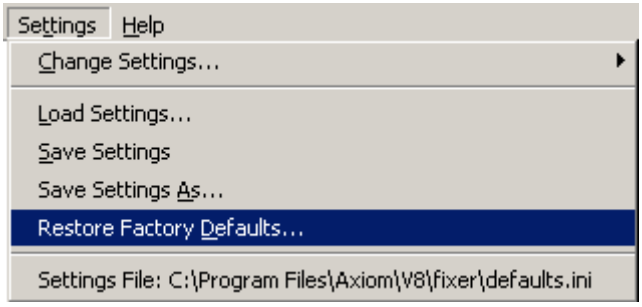
Factory Defaults

FileFixer has many settings that you can change to fine-tune the way it processes your files. Factory Defaults will work in most cases, so we won't cover *FileFixer's* settings in detail in this Quick Start.

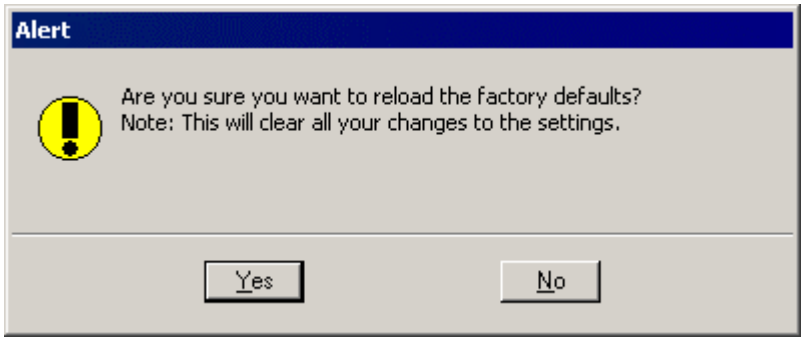
However, just in case someone experimented with *FileFixer's* settings before you began this Quick Start, let's reset *FileFixer* to Factory Defaults.

Here's your first assignment:

1. Select *FileFixer's* "Settings" menu, then select "Restore Factory Defaults…."

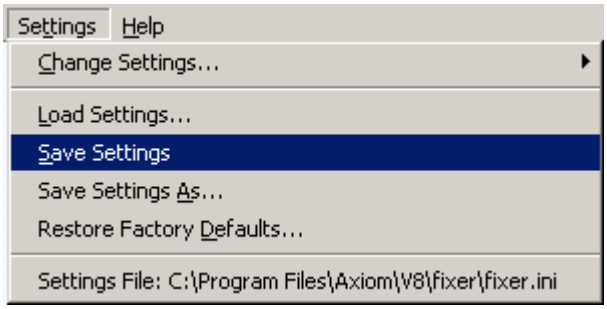


A confirmation dialog box will display:



2. Click the <Yes> button, this restores the Factory Defaults.

3. Select *FileFixer's* "Settings" menu and then select "Save Settings":



Selecting this saves the Factory Defaults to `fixer.ini`.

Note: *FileFixer* saves your search and repair settings in `.ini` files ("ini" is short for "initialization"). They are in ASCII format and can be modified with a text editor.

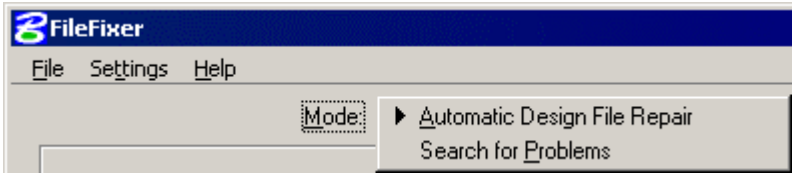
4. Next, we will exit *FileFixer* to prepare for the next Quick Start step, click the <Close> button on the lower right of *FileFixer's* main dialog box.

The remainder of this Quick Start requires Factory Defaults.

As you learn more about *FileFixer,* you can tweak its various options and save them using the "File" menu as illustrated above.

Quick Start Step 2 — Automatic Design File Repair

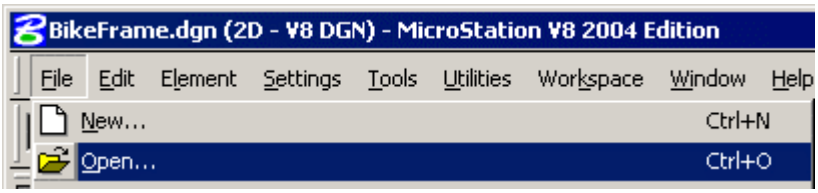
Use *FileFixer's* "Automatic Design File Repair" mode whenever you experience odd problems with your design file or cell library.



Let's walk through the steps:

1. Use MicroStation's "File" menu to open the `EmptyText.dgn` design file that is delivered with *FileFixer*. The standard installation location for this design file is:

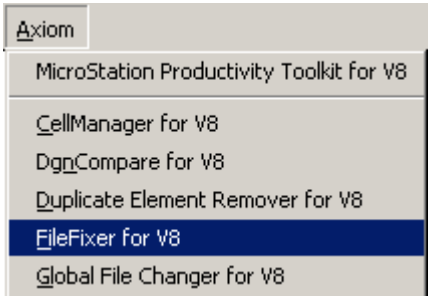
   `C:\Program Files\Axiom\v8\fixer\sample\EmptyText.dgn`


BikeFrame.dgn (2D - V8 DGN) - MicroStation V8 2004 Edition
File Edit Element Settings Tools Utilities Workspace Window Help
New... Ctrl+N
Open... Ctrl+O

   Even if *FileFixer* does not find any problems in the selected file, the following steps will nevertheless introduce you to the procedure for "Automatic Design File Repair".

   Note: Some of the sample files delivered with *FileFixer* are so corrupt that MicroStation cannot open them. If you happen to choose a file that MicroStation cannot open, just select another one to use for this step of the Quick Start.

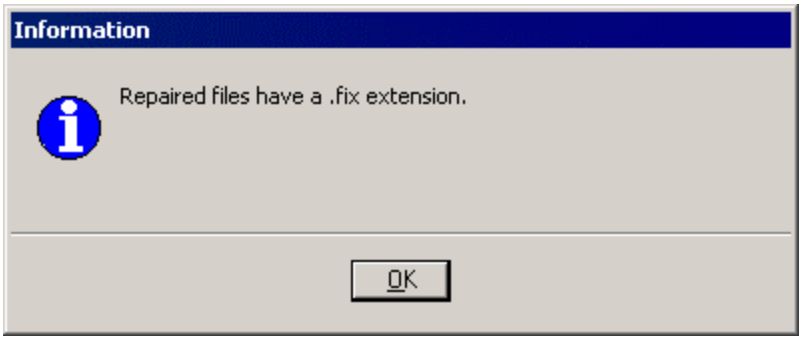
2. Select *FileFixer* from the "Axiom" menu.



3. The active design file appears in the "Which Files?" field by default. For example:

4. Press the <Start> button:

Once the processing is completed you will see the following message box:



It's that simple. But what just happened?

The file you selected was not modified — *FileFixer* created a copy. For example, if you selected `EmptyText.dgn`, *FileFixer* created a copy named `EmptyText.fix`.

FileFixer analyzed the copy (the ".fix" file) and performed the "Automatic Design File Repair" steps on it.

Note: The file "EmptyText.dgn" contains errors that are not handled by *FileFixer's* "Factory Default" settings. The reason for this is that some fixes are not turned "On" by default. This leads directly into the next Quick Start step, which covers how you use *FileFixer's* report and its error codes to handle situations like this.

FileFixer created a detailed report of its findings and repair actions (more about reports in the next Quick Start section).

You can open the repaired `.fix` file with MicroStation and get back to work! Use MicroStation's *File* | *Save As...* to rename the `.fix` copy (replacing the original file).

Congratulations — you have used *FileFixer* to perform automatic design file repair!

Quick Start Step 3 — *FileFixer's* Report

FileFixer's "Automatic Design File Repair" mode automatically repairs your files most of the time, so you won't always need to examine the report.

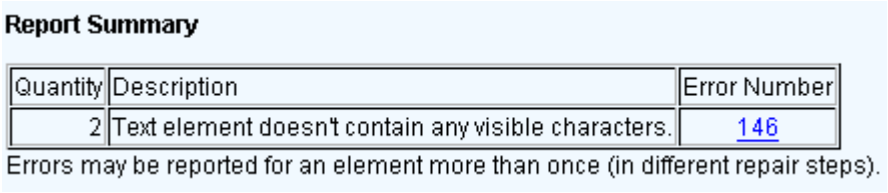
However, if a design file continues to misbehave after using "Automatic Design File Repair", the report will provide clues for fine-tuning *FileFixer's* options so that "Automatic Design File Repair" eradicates the problems.

Let's highlight the primary "landmarks" in a *FileFixer* report file.

1. Display the report created in the previous Quick Start step by pressing the <Display Report> button:

The report will open in your browser.

2. Find the “Report Summary” near the top of your report file. The “Report Summary” will look something like the following:



Report Summary

Quantity	Description	Error Number
2	Text element doesn't contain any visible characters.	146

Errors may be reported for an element more than once (in different repair steps).

Note: Depending on the sample files you selected, your “Report Summary” might just contain “No errors found”. The remainder of this Quick Start assumes that at least one problem is found in the selected practice file.

The number in the “Quantity” column of the “Report Summary” represents how many times the error was found. In this example, there are two instances of text, which doesn’t have any visible characters.

The “Description” column consists of a brief description of the error.

Finally, the “Error Number” column contains the specific *FileFixer* error code, which is also a hyper-link to *FileFixer’s* PDF documentation.

The higher the error code, the more severe the error (error “702” is more severe than “146”).

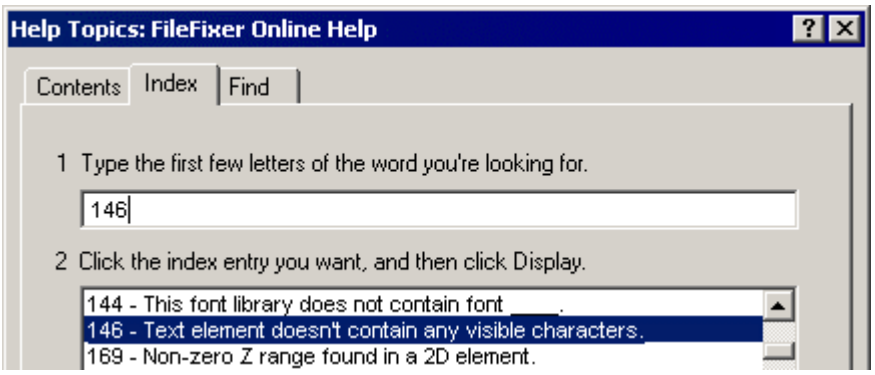
Note: If your file continues to misbehave after “Automatic Design File Repair”, it is an effective strategy to look more closely at the highest error codes first.

3. The format of the HTML report makes it easy to gain more information on the errors found. The error codes are links which, when clicked, will open the *FileFixer* PDF documentation, taking you to the section containing the error code. Place your cursor over the “146” and click your left mouse button.

   Adobe® Acrobat® Reader will jump to the appropriate passage in *FileFixer’s* User Guide.

4. You can also use *FileFixer’s* on-line help (Help | *Contents...)* and locate the section for this exact error code “146”:

Click the “Index” tab and in the first field, type in the error code “146”. You will see the entry for error code listed as “146 – Text element doesn’t contain any visible characters”. Double click on the listing for “146” error code in the second field and the full help text for this error will display in a window. In *FileFixer’s* “Help” file, all error codes are formatted this same way.


Help Topics: FileFixer Online Help
Contents
Index
Find
1 Type the first few letters of the word you're looking for.
146
2 Click the index entry you want, and then click Display.
144 - This font library does not contain font ____.
146 - Text element doesn't contain any visible characters.
169 - Non-zero Z range found in a 2D element.

That’s a brief introduction to *FileFixer* report files, the error code links to *FileFixer’s* PDF documentation and on-line help so you’re prepared to use them when the time comes to repair a file for real.

Quick Start Step 4 — Searching for Problems

Use *FileFixer’s* “Search for Problems” mode to detect hidden problems before they impact your projects or to confirm your files are “healthy” and problem-free.

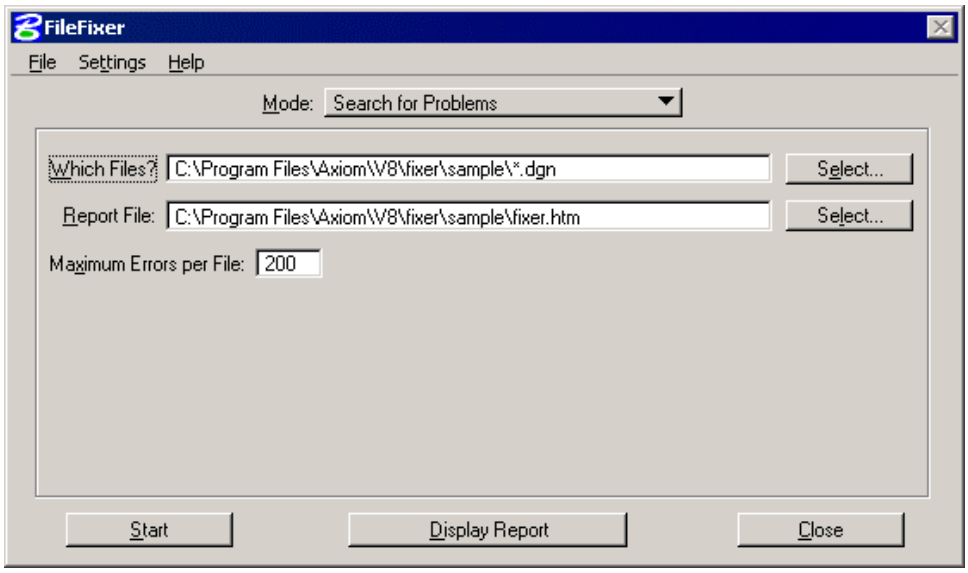
Other common uses for “Search for Problems” mode:

- Before submitting files to a client.
- Before accepting files from a sub-contractor.
- Before archiving a project.

Let’s get started.

1. Change *FileFixer’s* mode to “Search for Problems”:

2. In the “Which Files” field, enter the name of any design file that you want to analyze. You can also specify a wildcard, as in the following example.



The <Select> button offers a flexible way to select multiple files for processing. We are not going to delve into <Select> in this Quick Start — for more information, see the section titled “Choose Files to Process”.

3. To search for problems, press the <Start> button:

4. When processing is complete, click the <Display Report> button:

5. By default, *FileFixer* adds a seal-of-approval to files processed in “Search for Problems” mode:

Seal-of-approval was found:

License: Site license – Your Company.
Version: *FileFixer* 8.7f
Date: 19 May 2007
Time: 11:51:55 pm

1 -- Design file header is not on level 8. [911]
1 -- Class of element is illegal. [333]
1 -- Range of arc or ellipse is incorrect. [724]

End of seal-of-approval information.

The seal-of-approval records any problems found as in the above example. Additionally, no errors would be recorded in the seal-of-approval if none were found the last time the file was processed by *FileFixer* with “Search for Problems” mode.

If *FileFixer* reports errors, you know what to do — just follow the steps you learned earlier in Quick Start Step 2.

That’s all it takes to search for problems in any design file!

Follow the above procedure:

- Prior to submitting files to a client.
- Before accepting files from a contractor or sub-contractor.
- Periodically to perform preventative maintenance.
- Before archiving a project.
- Whenever a design file or cell library misbehaves.

Quick Start Step 5 — What’s Next?

Go ahead and try using *FileFixer* on your files. Remember, by default, *FileFixer* always backs up your original file, so you can experiment freely.

As you need to know more, use *FileFixer’s* help to learn more about the error messages reported.

Problem Element Viewer

FileFixer’s companion application, *Problem Element Viewer,* allows you to view the problem elements reported by *FileFixer.*

Problem Element Viewer also allows you to examine your files, element by element — even elements you cannot normally see using MicroStation alone.

Don’t hesitate to contact us if you have a question or need additional help.

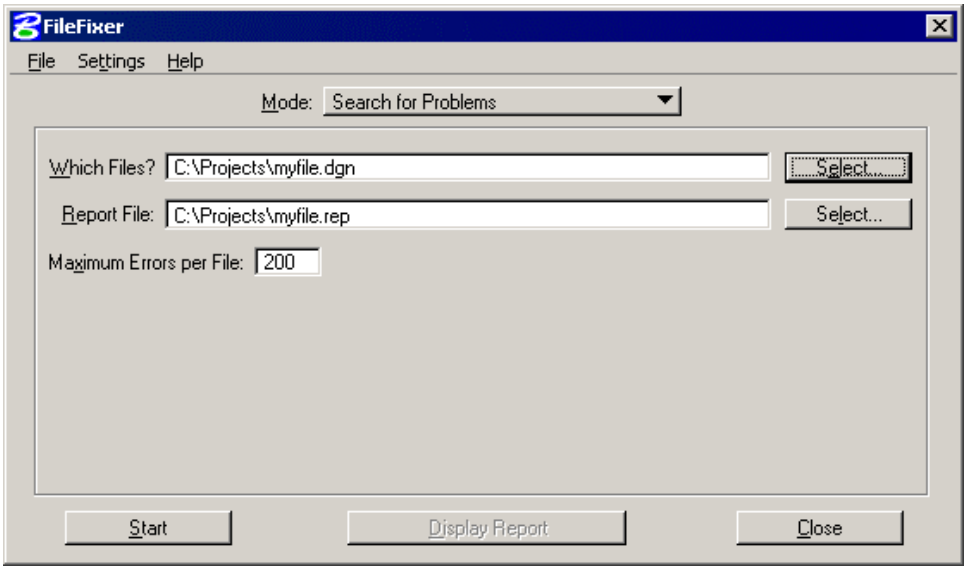
Chapter 5 — Running *FileFixer*

Getting started with *FileFixer*

Once *FileFixer* has been installed, select *FileFixer* from the Axiom menu on the MicroStation menu bar.

FileFixer is so easy to use you'll probably be able to diagnose and repair design files without reading the rest of this documentation. However, our clients confirm time after time that the more you know about *FileFixer* the more it can do for you.

The main dialog box



FileFixer's main dialog box.

Mode

To diagnose your files, set "Mode:" to "Search for Problems".

To repair your files, set "Mode:" "Automatic Design File Repair".

Which Files?

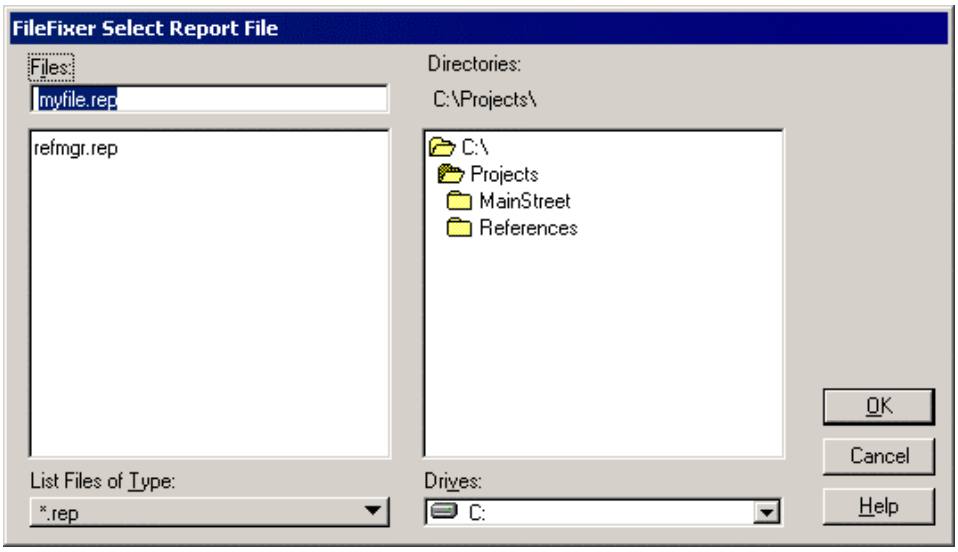
Tell *FileFixer* which file or files to process.

Example	Explanation
`C:\Project\MyFile.dgn`	A single file name.
	Select any assortment of files from one or more directories.
`C:\Project\*.dgn`	A wildcard expression — in this example, all design files in the C:\Project\ directory.
`C:\Project\*.dgn /S`	All design files in C:\Project\ and its sub-directories.
`/l C:\Project\MyFiles.txt`	MyFiles.txt is a file containing the names of files to be processed, one file name per line. The "/l" characters directs *FileFixer* to use the *contents* of the specified file.

Report File

Tell *FileFixer* where to save its report about the file or files processed.

The “Report File:” <Select...> button opens a dialog box that allows you to select a report file directory and name.



Tip: The *Settings* | *Change Settings...* | *Report File* dialog box gives you many selections for changing the default format and content of the report file.

<Start>

Press <Start> when you have selected all desired settings and are ready for *FileFixer* to process your files.

<Display Report>

Each report file contains *FileFixer*’s analysis of your design files and cell libraries, and what was done to remedy problems found.

Press <Display Report> to display a *FileFixer* report.

This button is disabled until you generate a report.

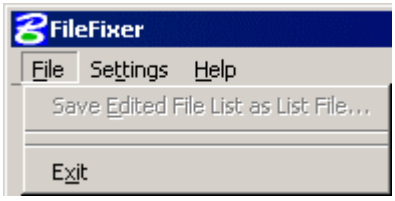
By default, text report files are displayed by “Notepad”, HTML report files are displayed by your default browser.

Use “Notepad” or your browser to print *FileFixer* report files.

Tip: Use *Settings* | *Change settings* | *Report File*... to switch between text and HTML report formats or to change the application which displays your report files.

Note: When you are creating more than one report file, the <Display Report> button displays the index file containing summaries from all reports.

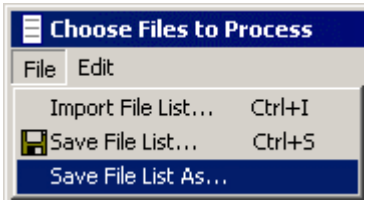
The File menu



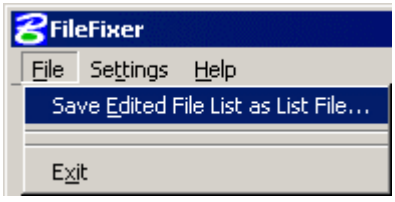
File | Save Edited File List as List File…

You can run *FileFixer* on a batch of files selected using <Select>.

If you want to save your file list for re-use later, use the *File* | *Save File List As*... function of the “Choose Files to Process” dialog as in the image below:



You can also select the “Save Edited File List as List File” option on the “File” pull down menu from *FileFixer’s* main dialog box:

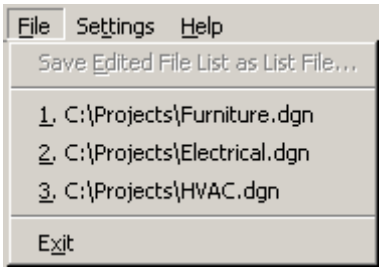


This option is enabled only when the main dialog’s “Which Files?” field contains “<List of 999 of 999 >”.

For much more information, see the “Choose Files to Process” section in this guide.

File History

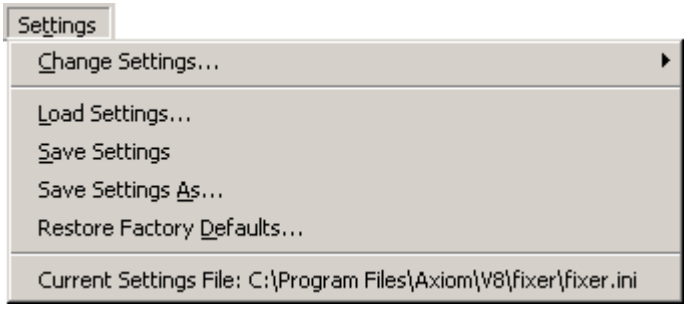
The “File” menu displays a history of files most recently processed. Files processed in batch do not appear in the file history list.



File | Exit

The “Exit” menu selection unloads *FileFixer*.

The Settings menu



The “Settings” menu.

Change Settings

“Change Settings” allows you to be able to change the General, Search, Repair, Report File and MicroStation Applications settings

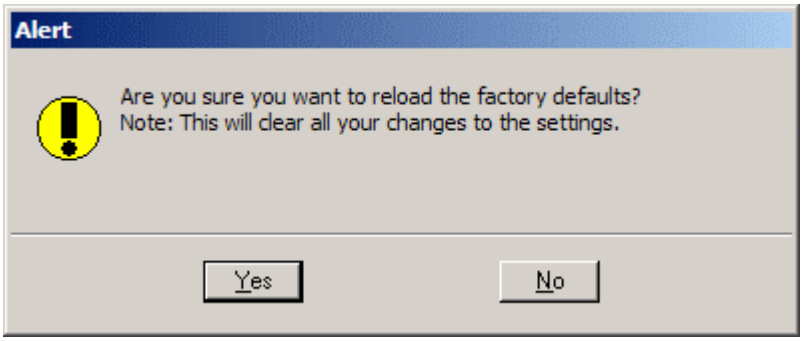
Load Settings...

When *FileFixer* is started up, the settings from the `fixer.ini` file are loaded. Use the “Load Settings” option to load settings from other .ini files that have been previously saved.

Use the `fixer.ini` to hold your default settings.

Restore Factory Defaults...

Select “Restore Factory Defaults” to reset all of *FileFixer’s* settings back to “factory default” values. These are the settings that your *FileFixer* came with when you first installed it. This is useful if you have been changing several settings and want to start fresh.



Save Settings

Use “Save Settings” to save all currently selected settings into the active settings file.

Save Settings As...

Use “Save Settings As...” to save your current selections with a file name of your choice.

Current Settings File:

General, report file, search and repair settings are saved in *FileFixer* “settings” files in ASCII format. The default settings file, `fixer.ini`, is created the first time you run *FileFixer*. It is saved in the same directory as `fixer.ma`.

You can save your settings file to a different file name and you can load previously saved settings.

Tip: Save your default settings to `fixer.ini`. These settings will load automatically when *FileFixer* loads.

The currently active “settings” file name is displayed at the bottom of the “Settings” menu.

FIXER_INI (custom location for your settings file)

The default location of the *FileFixer* settings file is the directory containing the fixer.ma.

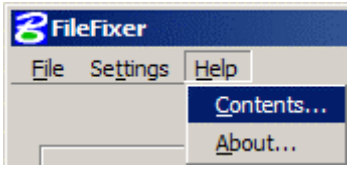
```
...axiom\V8\fixer\
```

You can also use a MicroStation configuration variable, FIXER_INI, to point *FileFixer* to a different directory containing the "settings" file of your choice. This is especially useful for groups of users running *FileFixer* from a shared location who want to define custom settings *locally*.

Just define the MicroStation configuration variable FIXER_INI and set it to the path of the directory containing your settings files. It will search that path for a `fixer.ini` file.

Tip: You can also change your default to a settings file of a different name by setting `FIXER_INI` to the full path and filename of a valid *FileFixer* "settings" file, such as "x:\users\home\bobsmith\BobsFixerSettings.ini".

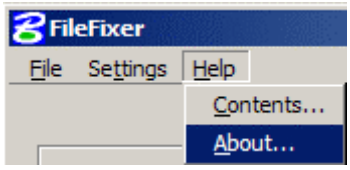
Help | Contents...



The *Help* | *Contents* option opens the full *FileFixer* User's Guide.

Tip: For a quick reference on any error that *FileFixer* reports, use *Help* | *Contents* to search for the error number.

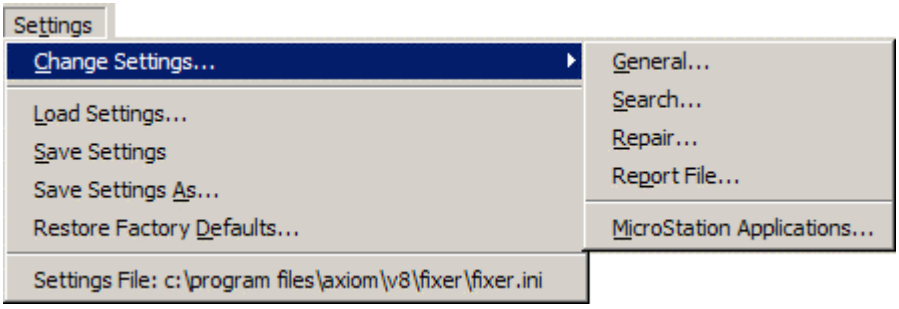
Help | About...



The *Help* | *About* option displays *FileFixer* version and licensing information.

Chapter 6 — The Change Settings Menu

Change Settings Menu



In most cases, you can successfully repair design files and cell libraries using *FileFixer's* default settings. The "Change Settings..." menu gives you access to *FileFixer's* special options for those cases that require a bit of extra tweaking.

Unless indicated otherwise, the settings illustrated in the following screen images reflect *FileFixer's* factory default settings.

General...

The "General ..." dialog box includes settings which aren't specifically related to search or repair.

One of these settings is used to tell *FileFixer* to repair the original design file and backup the original file, or to leave the original design file untouched and to create a fixed design file.

For more information, see General Settings.

Search...

The "Search" setting categories allow you to fine-tune which types of errors *FileFixer* looks for in your files and how they are reported in your report files.

For detailed descriptions, see the section titled "Search Settings".

Repair...

The “Repair” setting categories allow you to fine-tune how *FileFixer* repairs your files.

As you look at the report file generated by *FileFixer* during “Automatic Design File Repair” mode, you will see that each error noted has an identifying error number. You can use those numbers to find out more information about errors in the “Error Messages” section of this guide, including advice about which “Repair” settings might help repair your files.

A later section of this guide describes the “Repair” dialog box and its settings in detail.

Report File...

By default, *FileFixer* creates a single report file with information about all files processed. This default is suitable most of the time.

Over the years our clients have requested a variety of report settings, all of which are available to you.

You may find it convenient to set up a specific directory for *FileFixer* reports.

You may want individual reports for each design file or cell library with a master index that links to these reports.

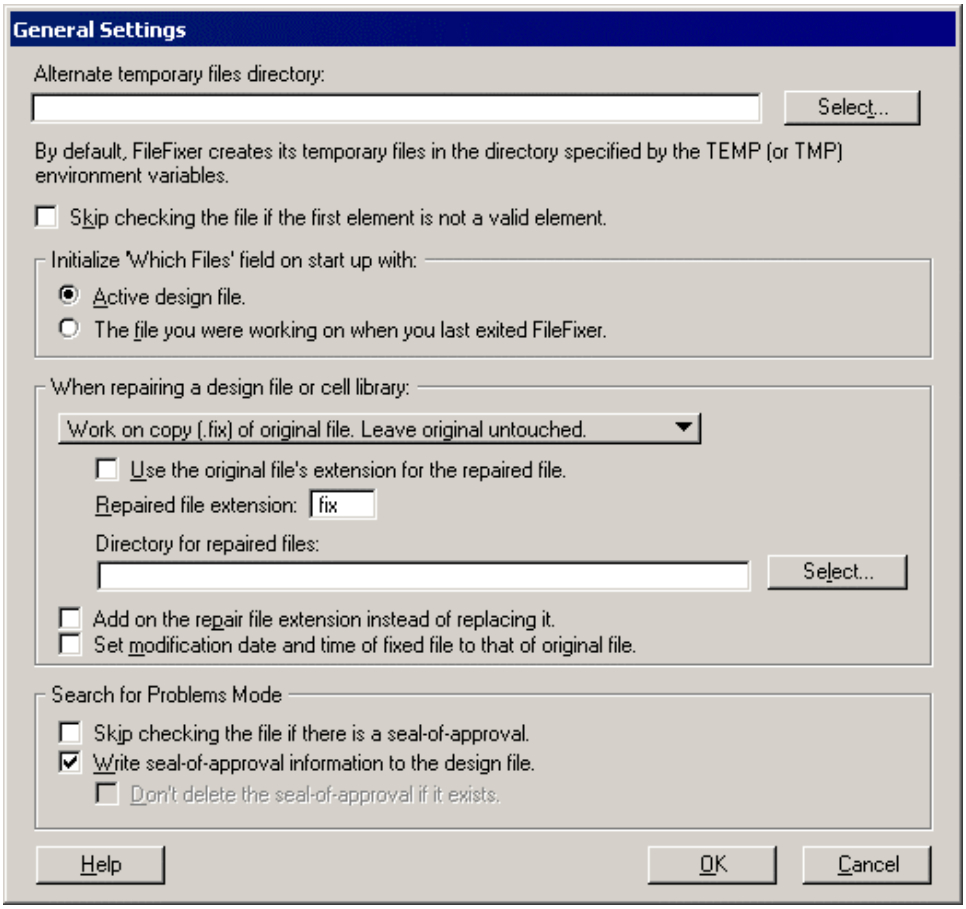
You may prefer a different extension or filename.

You may prefer to create a brief report with summary information only, or you may want to tailor report content in other ways.

The “Report File Settings” dialog box allows you to set up these scenarios, and you can preserve your preferences when you save *FileFixer* settings. A later section of this guide describes the “Report file” dialog box and its settings in detail.

Chapter 7 — General Settings

The General Settings Dialog Box


General Settings
Alternate temporary files directory:
Select...
By default, FileFixer creates its temporary files in the directory specified by the TEMP (or TMP) environment variables.
Skip checking the file if the first element is not a valid element.
Initialize 'Which Files' field on start up with:
Active design file.
The file you were working on when you last exited FileFixer.
When repairing a design file or cell library:
Work on copy (.fix) of original file. Leave original untouched.
Use the original file's extension for the repaired file.
Repaired file extension: fix
Directory for repaired files:
Select...
Add on the repair file extension instead of replacing it.
Set modification date and time of fixed file to that of original file.
Search for Problems Mode
Skip checking the file if there is a seal-of-approval.
Write seal-of-approval information to the design file.
Don't delete the seal-of-approval if it exists.
Help
OK
Cancel

"General Settings" dialog box.

Alternate temporary files directory

FileFixer creates a number of temporary files while it is repairing a design file or cell library.

FileFixer determines the location for these temporary files in the following manner:

Use the directory specified in the "Alternate temporary files directory" on the General Settings dialog box.

If no directory is specified in the "Alternate temporary files directory" field then use the directory specified by the Windows "TEMP" environment variable.

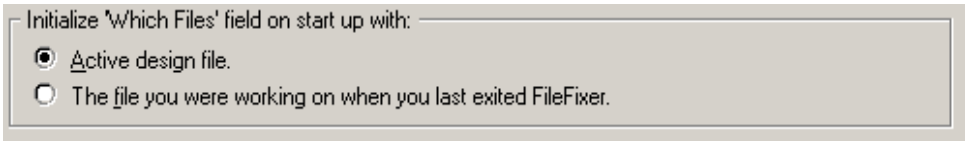
If the “TEMP” environment variable isn’t specified then use the directory specified by the Windows “TMP” environment variable.

If the “TMP” environment variable isn’t specified then put them in the directory containing the design file or cell library being processed.

Skip checking the file if the first element is not a valid element.

If the first element in a design file is not a design file header then skip processing this file. This option can be used to skip files that are not V8 design files or cell libraries.

Initialize “Which Files” field on startup with:


Initialize 'Which Files' field on start up with:
Active design file.
The file you were working on when you last exited FileFixer.

Use these settings to specify how the “Which Files” field is initialized when *FileFixer* is started.

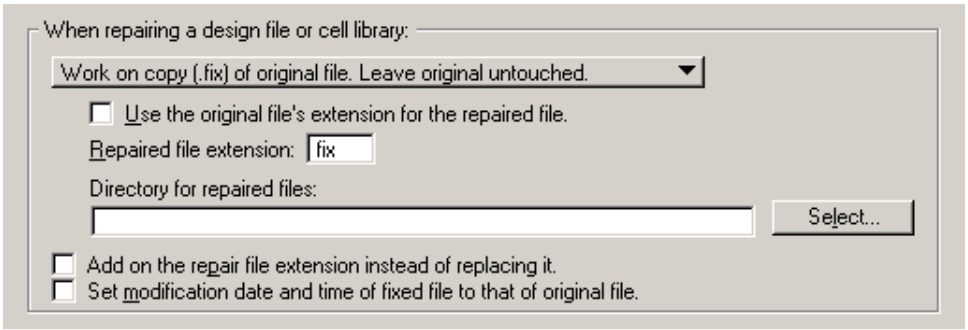
Use the “Currently loaded design file” setting when you want *FileFixer* to be initialized with the design file currently loaded in MicroStation.

Use the “File name that was in the field when last terminated” setting when you want *FileFixer* to remember the name of the file you were working with when you last quit *FileFixer*.

Note: When you have a demonstration license, *FileFixer* always initializes the “Which files” field with the sample files in ...\Axiom\Fixer\v8\sample\.

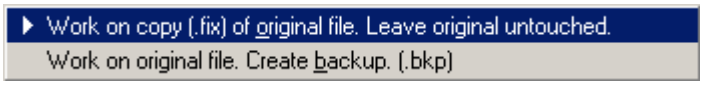
You can process *any* V8 design file or cell library with a demonstration license.

When repairing a design file or cell library


When repairing a design file or cell library:
Work on copy (.fix) of original file. Leave original untouched.
Use the original file's extension for the repaired file.
Repaired file extension:
fix
Directory for repaired files:
Select...
Add on the repair file extension instead of replacing it.
Set modification date and time of fixed file to that of original file.

“When repairing a design file or cell library:” settings.

Backup/repaired file name


Work on copy (.fix) of original file. Leave original untouched.
Work on original file. Create backup. (.bkp)
Repaired file extension:
fix

FileFixer offers the option to work on the *original* design file (after first backing up the design file) or to work on a *copy* of the original design file (leaving the original unchanged). With either option you have the ability to specify the final extension and directory where the backup file or repaired file will be placed.

Demo Users Note: We do *not* recommend selecting "Work on original" when you have a demonstration license. If your file is greater than 100KB, a backup will be created, but a repaired file will not be created to replace the original file.

The backup file extension is limited to three characters on all platforms.

The "?" character provides some flexibility when naming backup file names. In this context it is *not* a traditional wildcard character.

One or more question marks "?" can be used to create a numerical extension which automatically increments.

You can use up to three question marks in the extension "???", which supports up to 999 backup files. If you attempt to make the 1000th backup, *FileFixer* will notify you that no more backup file extensions are available (until you select a different extension, different output directory or delete the previous backups).

The question marks must be at the end of the extension. This means that the extension "b?k" is not recognized.

Examples:

Original file name	Backup or repaired filename expression	Backup or repaired filename created
Axiom.ff1 Axiom.ff2 ... Axiom.ff9	bk?	Axiom.bk1 Axiom.bk2 ... Axiom.bk9
Axiom.f01 Axiom.f02 ... Axiom.f99	b??	Axiom.b01 Axiom.b02 ... Axiom.b99
Axiom.dgn	???	Repeated *FileFixer* processing of the same file creates: Axiom.001 Axiom.002 Axiom.003

Use the original file's extension for the repaired file.

Select this option if you want your repaired file to retain the extension of the original file. Of course, *FileFixer* requires you to specify a different output directory for the repaired file (so your original is always retained).

Add on the repair file extension instead of replacing it.

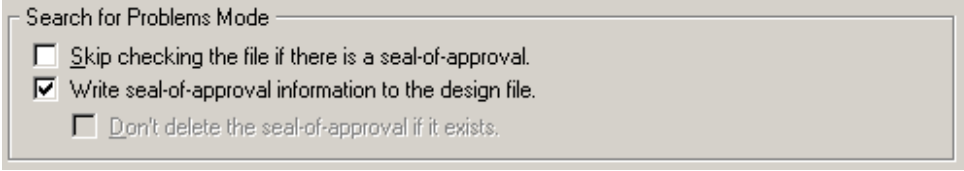
Using this repair setting the repair file extension will be added onto the end of the design file or cell library file name instead of replacing the extension. For example, when creating a fix file, the repaired file extension ".fix" will be added onto the end of the design file "axiom.dgn" making the fixed file name "axiom.dgn.fix".

Set modification date and time of fixed file to that of the original file.

Use this setting when you want the date and time stamp of the fixed file to be the same as the design file or cell library being repaired. Normally, the date and time of the fixed file will be the date and time the file was repaired.

When you are working on the original design file and creating a backup file the backup file will always maintain the date and time stamp of the original file. Only the original file's date and time stamp will be optionally kept or set to the current date and time.

Search for Problems Mode


Search for Problems Mode
Skip checking the file if there is a seal-of-approval.
Write seal-of-approval information to the design file.
Don't delete the seal-of-approval if it exists.

"Search for Problems Mode" settings.

Skip checking the file if there is a seal-of-approval.

If the seal-of-approval is found and there are no error numbers greater than 199 in it then *FileFixer* will not process this design file.

Write seal-of-approval information to the design file.

FileFixer has the ability to add the seal-of-approval information to your design file when running in "Search for Problems" mode. The seal-of-approval information contains the error numbers found, date and time when *FileFixer* analyzed the file and the level of error reporting that was done (e.g. "Fatal error only", "Errors and fatal errors", etc.).

If you elect to de-select this setting, you can also direct *FileFixer* to retain or delete any pre-existing seal-of-approval:

Don't delete the seal-of-approval if it exists.

Note: If the design file's date is newer than the date saved in the seal-of-approval then the message "WARNING: File modification date exceeds certification date" is output to the report file.

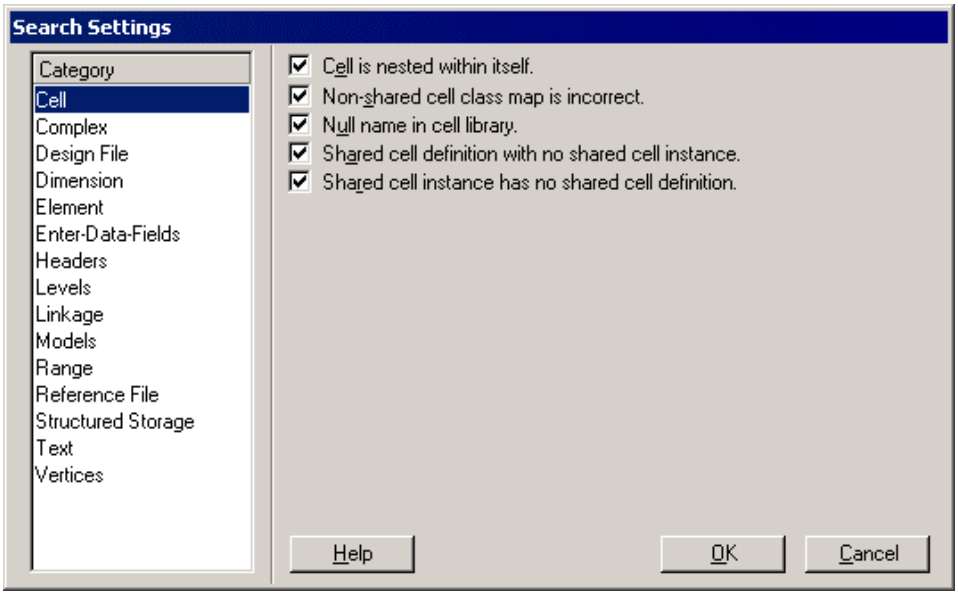
Note: The seal-of-approval is stored in a type 66 element saved in the non-model section of a V8 design file or cell library.

Chapter 8 — Search Settings

Factory defaults

Unless indicated otherwise, the settings illustrated in the following screen images reflect *FileFixer's* factory default settings.

Cell



"Search Settings", "Cell" Category

Cell is nested within itself.

Use this setting to locate duplicate nested cells.

The repair setting that corrects this problem is *Settings* | *Change Settings* | *Repair...* | *Cell* | *Delete cell nested within itself.*

The error number that is reported when the nested cell is in a design file is 383.

Non-shared cell class map is incorrect.

Error number 717 is reported when the classes stored in the cell header don't include all of the actual classes used by components in the cell.

The repair setting that corrects this error is *Settings* | *Change Settings* | *Repair...* | *Cell* | *Adjust non-shared cell class map*. It is "On" by default.

Null name in cell library.

A type 2 cell with a null (empty) cell name is found in a cell library.

The repair setting that corrects this error is *Settings* | *Change Settings* | *Repair...* | *Cell* | *Correct null cell names (type 2s only) by setting the H-bit.*

The error number reported is 570.

Shared cell definition with no shared cell instance.

A shared cell definition has been found and there is no shared cell instance that uses this definition. This happens quite often when converting DWG to V8.

The repair setting that corrects this error is *Settings* | *Change Settings* | *Repair...* | *Cell* | *Delete shared cell definition with no shared cell instance.*

The error number reported is 303.

Shared cell instance has no shared cell definition.

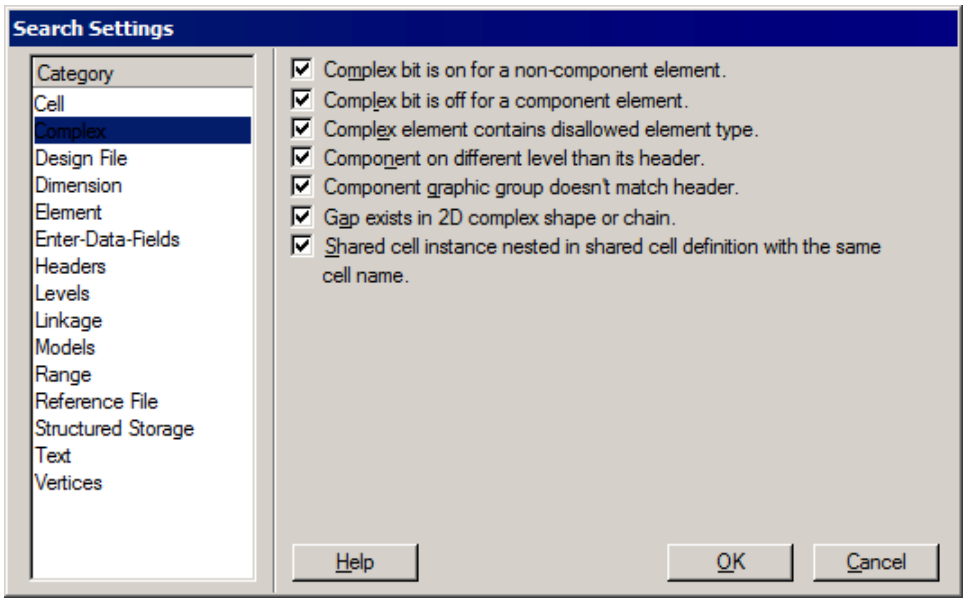
A shared cell instance was found with no corresponding shared cell definition.

The repair setting that corrects this error is *Settings* | *Change Settings* | *Repair...* | *Cell* | *Create dummy shared cell definition for missing definitions.*

This setting will create a dummy shared cell definition. What this setting does is create a shared cell definition element with a text element as its only component. The text element contains the name of the shared cell. This will graphically show you the locations where the cell is used..

The error number reported is 768.

Complex



"Search Settings", "Complex" category.

Complex bit is on for a non-component element.

An element that is not a component of a complex element has its complex bit set.

The repair setting that corrects this error is *Settings* | *Change Settings* | *Repair...* | *Complex* | *Correct complex bit errors*.

The error number reported is 702.

Complex bit is off for a component element.

An element that is a component of a complex element does not have its complex bit set.

The repair setting that corrects this error is *Settings* | *Change Settings* | *Repair...* | *Complex* | *Correct complex bit errors*.

The error number reported is 703.

Complex element contains disallowed element type.

The following error numbers are controlled by this search setting:

Error number **323** — Complex elements should not contain deleted elements.

Error number **721** — Each type of complex element (like cells and text nodes) can only contain certain types of elements. For example, a text node (type 7) can contain only text elements (type 17s). A connected string (type 12) can contain only linear elements such as lines (type 3s), line strings (type 4s), arcs (type 16s), etc. The reported element cannot legally be contained in the parent complex element (identified in the error message).

The repair setting that corrects error number 721 is *Settings* | *Change Settings* | *Repair...* | *Complex* |*Handle illegal components in complex header*.

Component on different level than its header.

The indicated element is part of a complex element and is on a different level than its parent element.

The repair setting that corrects this error is *Settings* | *Change Settings* | *Repair...* | *Complex* | *Move components to same level as complex header*.

The error number reported is 336.

Component graphic group doesn’t match header.

All component elements must have the same graphic group number as the complex header.

This error is not reported on component elements that are part of regular cells (type 2), shared cell definitions (type 34) and raster headers (type 87 or type 90).

The repair setting that corrects this error is *Settings* | *Change Settings* | *Repair...* | *Complex* | *Set graphic group of component to match its header*.

The error number reported is 341.

Gap exists in 2D complex shape or chain.

When creating a 2D complex shape or chain from multiple linear elements that have gaps between the endpoints, MicroStation will not actually create the connecting lines, but MicroStation will *display* a line between those endpoints. If these complex elements are later dropped, gaps will exist between the component elements.

The repair setting that corrects this error is *Settings* | *Change Settings* | *Repair...* | *Complex* | *Add line to bridge gap in 2D complex shape or chain*. The repair setting is off by default.

The error number reported is 151.

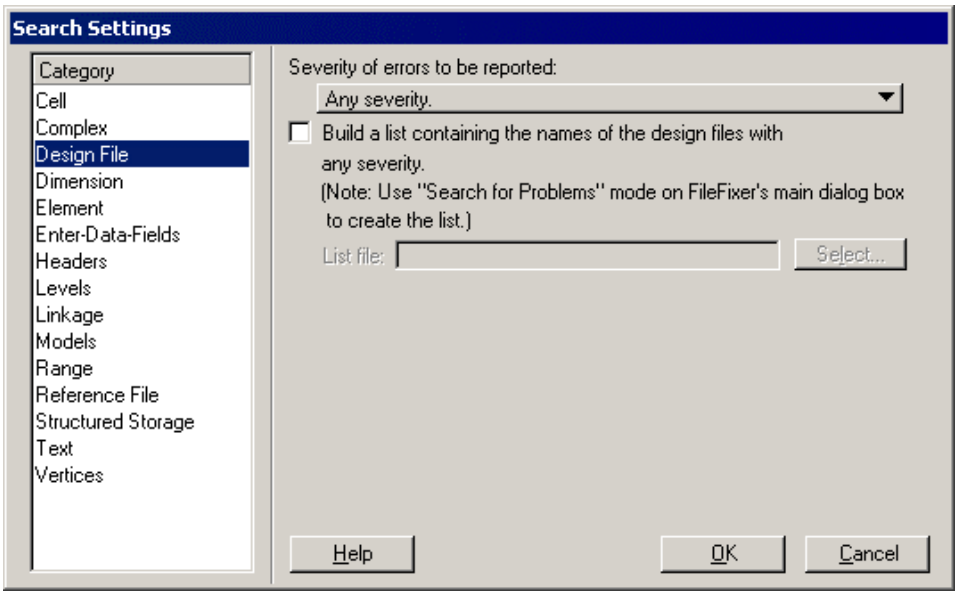
Shared cell instance nested in a shared cell definition with the same cell name.

A shared cell instance cannot be part of the shared cell definition that defines the shared cell instance.

The repair setting that corrects this error is *Settings* | *Change Settings* | *Repair...* | *Complex* | *Delete shared cell instance nested in a shared cell definition with the same name.*

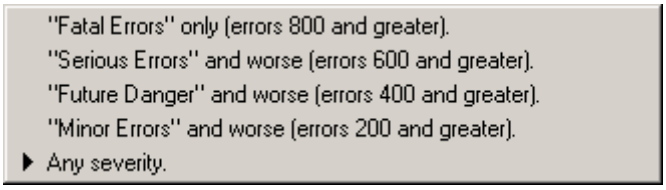
The error number reported is 733.

Design File



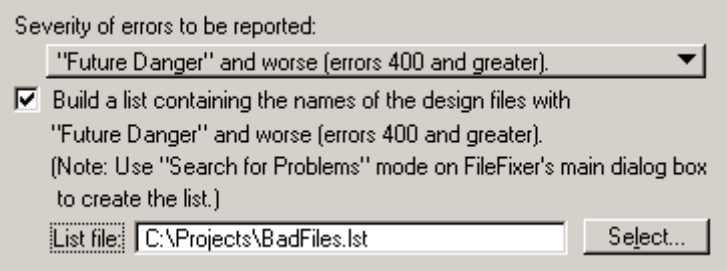
"Search Settings", "Design File" category.

Severity of errors to be reported:


"Fatal Errors" only (errors 800 and greater).
"Serious Errors" and worse (errors 600 and greater).
"Future Danger" and worse (errors 400 and greater).
"Minor Errors" and worse (errors 200 and greater).
Any severity.

This search setting gives you the ability to search and report only errors in a user-selectable severity range. "Any severity" is the default.

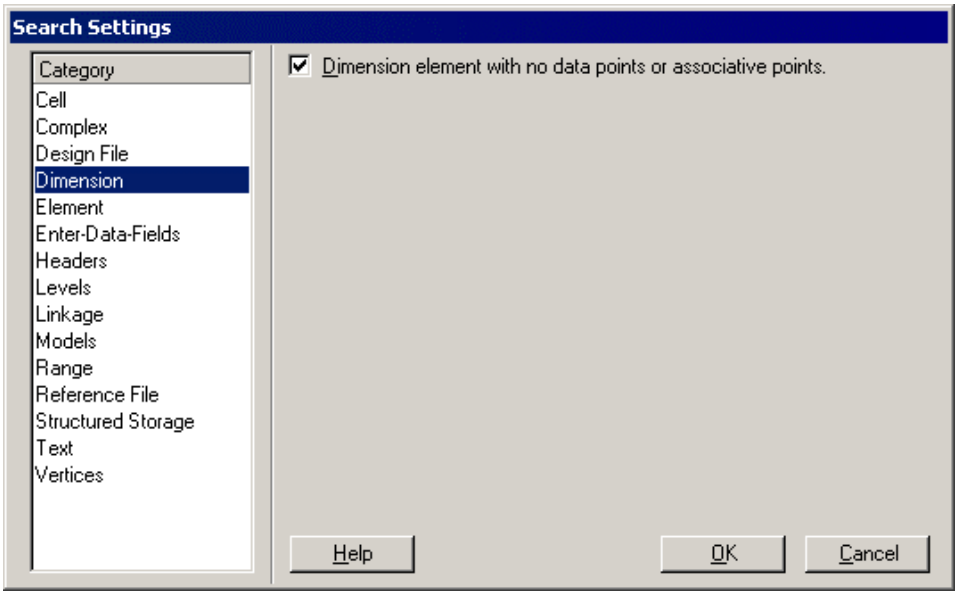
Build a list containing the names of the design files with ...


Severity of errors to be reported:
"Future Danger" and worse (errors 400 and greater).
Build a list containing the names of the design files with
"Future Danger" and worse (errors 400 and greater).
(Note: Use "Search for Problems" mode on FileFixer's main dialog box
to create the list.)
List file:
C:\Projects\BadFiles.lst
Select...

FileFixer can automatically build a list of files that manifest the error severity you select.

The list will only be generated in Search for Problems mode.

Dimension



“Search Settings”, “Dimension” category.

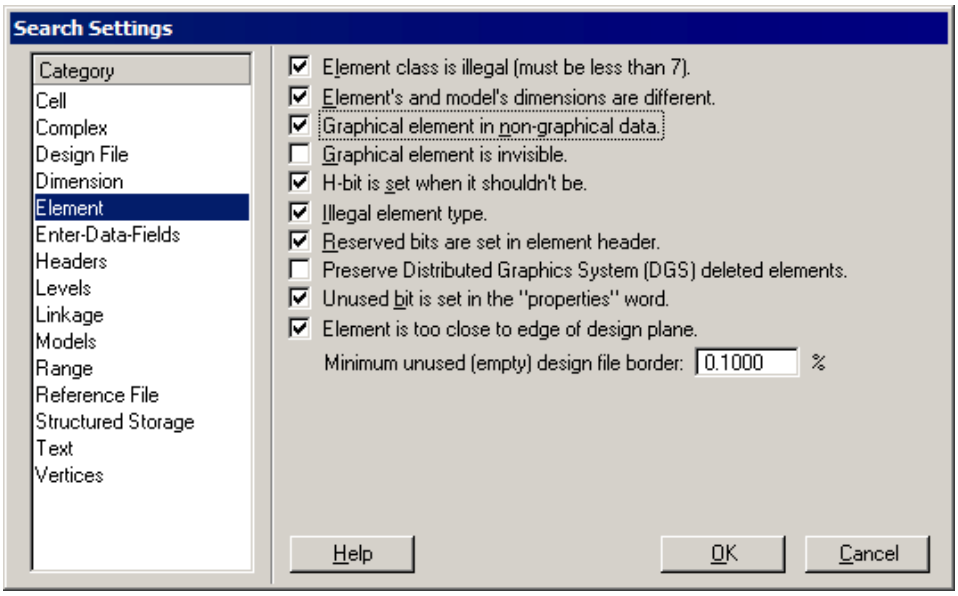
Dimension element with no data points or associative points.

A dimension element must have at least one data or associative point. An associative point is a point that ties this dimension element to another element in the design file.

The repair setting that corrects this error is *Settings* | *Change Settings* | *Repair...* | *Dimension* | *Remove dimension element with no data points or associative points*.

The error number reported is 722.

Element



"Search Settings", "Element" category.

Element class is illegal (must be less than 7).

The class of this element is undefined (greater than 6).

The repair setting that corrects this error is *Settings* | *Change Settings* | *Repair...* | *Element* | *Set illegal class to* ___.

The error number reported is 333.

Element's and model's dimensions are different.

The element's dimension is different that the model's dimension. For example, a 3D line in a 2D model. Such elements cannot always be modified without first repairing the element's dimension.

The repair setting that corrects this error is *Settings* | *Change Settings* | *Repair...* | *Element* | *Set the element's dimension to match the model's dimension*.

The error number reported is 437.

Graphical element in non-graphical data.

Graphical elements in the non-graphical data area can cause MicroStation V8i to lock up or abort. This data area is for reference file attachments, application data elements, etc.

The repair setting that corrects this error is *Settings* | *Change Settings* | *Repair...* | *Element* | *Delete graphical element in non-graphical data.*

The error number reported is 819.

Graphical element is invisible.

MicroStation V8 has the ability to make an element “invisible”. It will not be displayed in MicroStation’s GUI or be plotted. This is a new MicroStation feature so these elements are not really corrupted, which is why this setting is OFF by default.

If you use Cimmetry’s AutoVue application, which does display these elements, then the AutoVue GUI can be a little confusing. That is the reason for this setting.

There is a new repair setting named “Fix invisible graphical element by” that allows you to either make these elements visible or to delete them.

The error number reported is 240.

H-bit is set when it shouldn’t be.

Turn off the hole bit (H-bit) of this element.

The search setting that detects this error is *Settings* | *Change Settings* | *Search...* | *Element* | *H-bit is set when it shouldn’t be.*

The error number reported is 304.

Illegal element type.

FileFixer detected illegal element types, such as type 0, 1, 20 and 127.

The repair setting that corrects these errors is *Settings* | *Change Settings* | *Repair...* | *Element* | *Remove illegal element types* (ON by default)

The error number reported is 380.

Reserved bits are set in element header.

Bits 10:15 are unused in the second word of every MicroStation V8 element. This allows for future enhancements of the element format. These bits should be clear (set to 0). If they are not, this message is generated.

The repair setting that corrects this error is *Settings* | *Change Settings* | *Repair...* | *Element* | *Fix reserved bits by clearing them*.

The error number reported is 103.

Preserve Distributed Graphics System (DGS) deleted elements.

There is a special type of attribute linkage used by an old Intergraph application (Distributed Graphics System) on deleted elements. When this setting is selected, *FileFixer* treats these deleted elements as though they were active (non-deleted) elements.

Unused bit is set in the “properties” word.

There is an unused bit in the “properties” word of each graphic element. The information saved in the “properties” word is the element’s class and the element’s properties (e.g. locked, hole). It is intended to allow future enhancement of the MicroStation element format, but was never used. This bit should be clear (set to 0). If it is not, this message is generated. Technically this is not an error because there are no known programs that rely on this bit in any way.

The error number reported is 132.

Element is beyond the edge of the design plane.

This element is located dangerously close to the edge of the design plane (or design cube) — or *beyond* the edge.

FileFixer usually determines this error by checking the element’s range values.

For ellipses, arcs, text and text nodes, *FileFixer* checks the element’s origin.

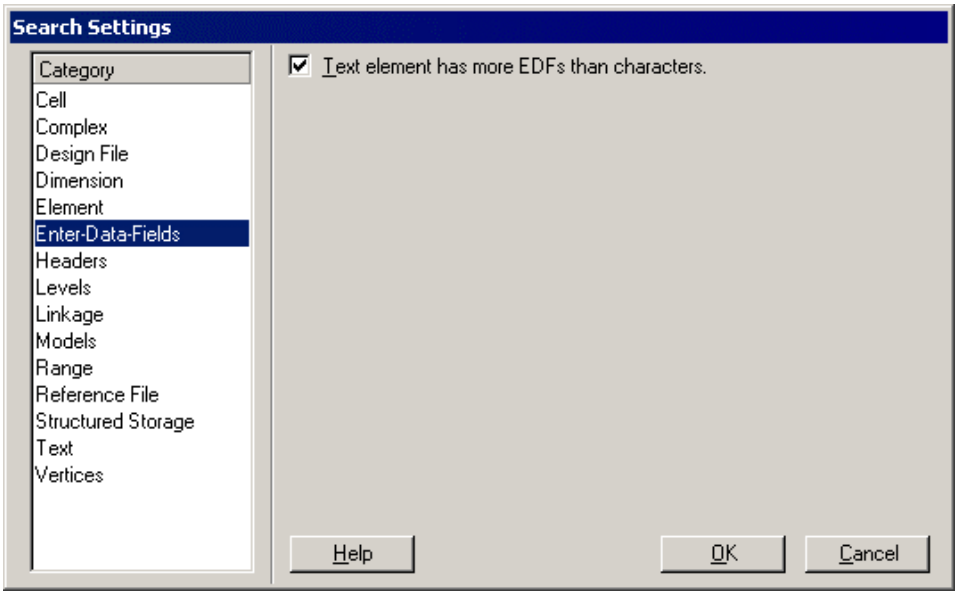
For curves, *FileFixer* checks the beginning and ending slope points.

The error number reported is 119 when the element is dangerously close to the edge of the design plane.

The error number reported is 619 when at least one element coordinate is *beyond* the edge of the design plane.

Note: V8 element coordinates range from 4,503,599,627,370,495 (max) to -4,503,599,627,370,496 (min). [From …\Program\MicroStation\mdl\include\msdefs.h.]

Enter-Data-Fields



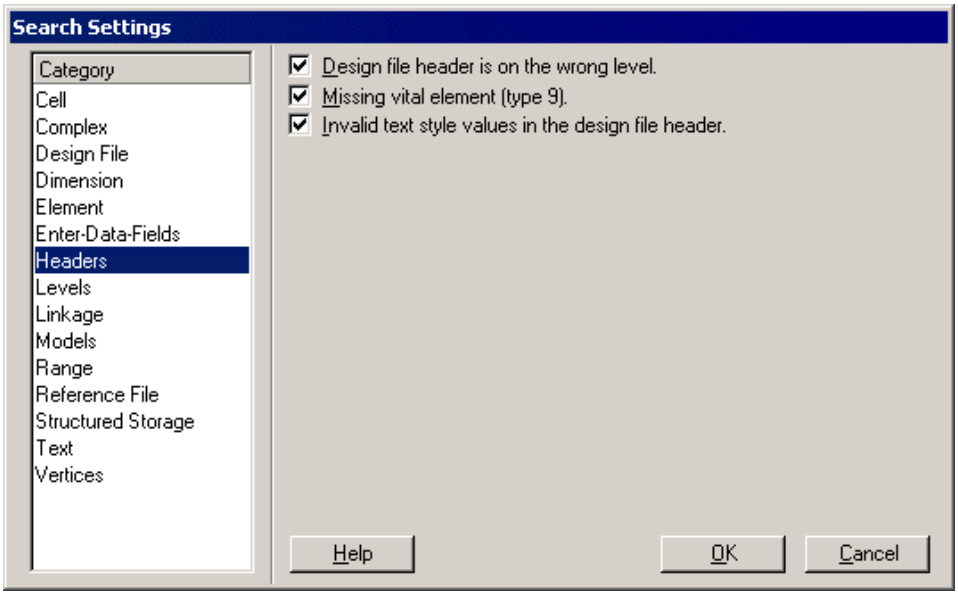
“Search Settings”, “Enter-Data-Fields” category.

Text element has more EDFs than characters.

A text element with more EDFs (Enter-Data-Fields) than characters has been encountered.

The error number reported is 191.

Headers



"Search Settings", "Headers" category.

Design file header is on the wrong level.

The design file header (type 9) element is on the wrong level. It *must* be on level number 8. When it is not then MicroStation cannot open the design file.

The repair setting that corrects this error is *Settings* | *Change Settings* | *Repair...* | *Element* | *Fix reserved bits by clearing them.*

The error number reported is 911.

Missing vital element (type 9).

Error number 902 is reported when the first non-model element (type 9 design file header) in a design file or cell library has been deleted or is badly damaged.

Invalid text style values in the design file header.

The default text style is saved in each design file's TCB and is used by default when no other text style is defined and active. When the default text style settings become

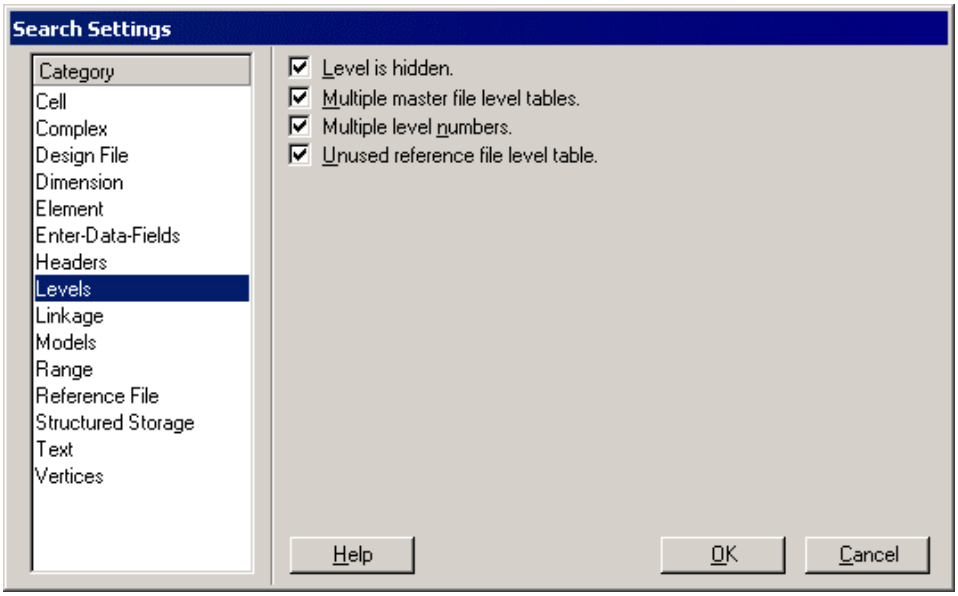
corrupted, merely attempting to place text (with default text style) may cause MicroStation to abort.

FileFixer detects default text style corruption by testing for illegal values.

The error number reported is 745.

When *FileFixer* detects and reports this form of corruption, MicroStation users can enter legal values by selecting *Element* | *Text Styles* from MicroStation's main menu, then the "Advanced" tab.

Levels



"Search Settings", "Levels" category.

Level is hidden.

FileFixer detected a hidden level.

The error number reported is 444.

The repair setting that corrects this error is *Settings* | *Change Settings* | *Repair...* | *Levels* | *Display hidden levels* (OFF by default).

Multiple master file level tables.

FileFixer detected more that one master file level table.

The error number reported is 630.

The repair setting that corrects this error is *Settings* | *Change Settings* | *Repair...* | *Levels* | *Delete unused master file level tables* (OFF by default).

Multiple level numbers.

FileFixer detected more than one level with the same level number.

The repair setting that corrects this error is *Settings* | *Change Settings* | *Search...* | *Levels* | *Delete the duplicate level number*.

The error number is 429.

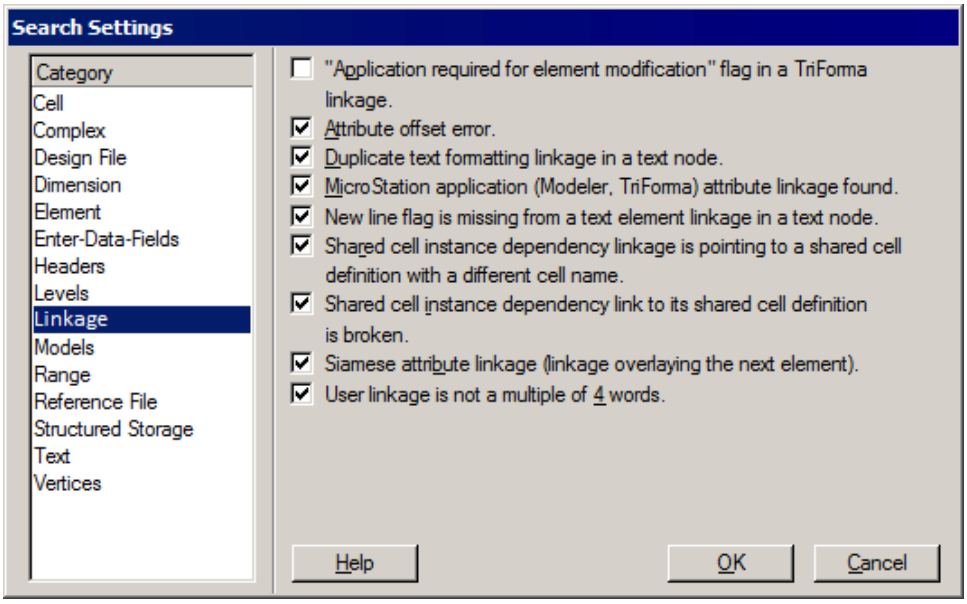
Unused reference file level table.

FileFixer found a reference file level table that has no attached reference file using it.

The error number reported is 267.

The repair setting that corrects this error is *Settings* | *Change Settings* | *Repair...* | *Levels* | *Delete unused reference file level tables* (OFF by default).

Linkage



“Search Settings”, “Linkage” category.

“Application required for element modification” flag in a TriForma linkage.

Elements, created by TriForma, can be set to prevent MicroStation, when run without TriForma loaded, from allowing a user to modify them. This is done by setting the “application required for element modification” flag in the TriForma linkage.

This is not a corruption, but it is a potentially unwanted condition that can be fixed with *FileFixer*. As a result this search setting is off by default.

The error number reported is 235.

The repair setting that fixes this error is *Settings* | *Change Settings* | *Repair...* | *Linkage* | *Clear "application required for element modification" flag in a TriForma linkage*.

Attribute offset error.

The element's attribute offset was not what *FileFixer* expected. This may indicate the linkage area of the element is damaged.

The error number reported is 514.

Duplicate text formatting linkage in a text node.

Normally line feeds are maintained between text strings when editing a text node. When more than one text formatting linkage exists in each of the text elements in a text node, and the line feed flag is turned off in one or more of those linkages, editing the text node will unexpectedly strip the line feeds from the text node.

Repair with *FileFixer's* "Automatic Design File Repair" command with the *Settings* | *Change Settings* | *Repair...* | *Linkage* | *Delete duplicate text formatting linkage in a text node* setting turned on.

The error number reported is 545.

MicroStation application-specific attribute linkage found.

As *FileFixer for V8* evolves, various application-specific settings may be added. As of this writing, TriForma-specific settings are supported. When the first element is detected which contains TriForma attribute linkage, *FileFixer* automatically activates TriForma-specific settings when processing the file.

FileFixer outputs a message when the first application-specific element is detected For example:

```
4000000, ID=43676 -- This is the first element indicating this file contains
TriForma attribute linkage.  [62].
```

Application-specific settings are displayed on the *Settings* | *Change Settings* | *MicroStation Applications* dialog.

The error number reported is 62.

New line flag is missing from a text element linkage in a text node.

Normally there is a line feed between text strings in a text node. When this error is reported the user attribute linkage for this component text element doesn't contain the expected new line attribute flag. Editing a text node with this problem will result in the text node unexpectedly losing its line feeds.

Repair with *FileFixer's* "Automatic Design File Repair" command with the *Settings | Change Settings | Repair... | Linkage | Add new line flag missing in a text element linkage that is part of a text node* setting turned on.

The error number reported is 544.

Shared cell instance dependency linkage is pointing to a shared cell definition with a different cell name.

Some MicroStation add-on applications create shared cell instances that use a dependency linkage to locate their definition rather than a cell name. Use this Search setting to report a shared cell instance whose dependency linkage contains an element ID that points to a shared cell definition with a different cell name than the instance.

Repair with *FileFixer's* "Automatic Design File Repair" command with the *Settings Change Settings | Repair… | Change the name of the shared cell instance, with dependency linkage, to match the shared cell definition's name if they don't match.*

The error number reported is 168.

Shared cell instance dependency link to its shared cell definition is broken.

Some MicroStation add-on applications create shared cell instances that use a dependency linkage to locate their definition rather than a cell name. Use this Search setting to report a shared cell instance whose dependency linkage contains an element ID that is broken because it doesn't match the ID of any available shared cell definitions.

Repair with *FileFixer's* "Automatic Design File Repair" command with the *Settings Change Settings | Repair… | Set the shared cell instance dependency linkage's element ID to the shared cell definition's element ID.*

The error number reported is 767.

Siamese attribute linkage (linkage overlaying the next element).

Use this Search setting to detect when an element's attribute linkage extends beyond the end of the element. That means the linkage overlaps the *next* element in the file — applications which depend upon data in the linkage area could malfunction or crash. Accessing a database using this linkage could fail.

Repair with *FileFixer's* "Automatic Design File Repair" command with the *Settings | Change Settings | Repair... | Linkage | Remove siamese linkage* setting turned on.

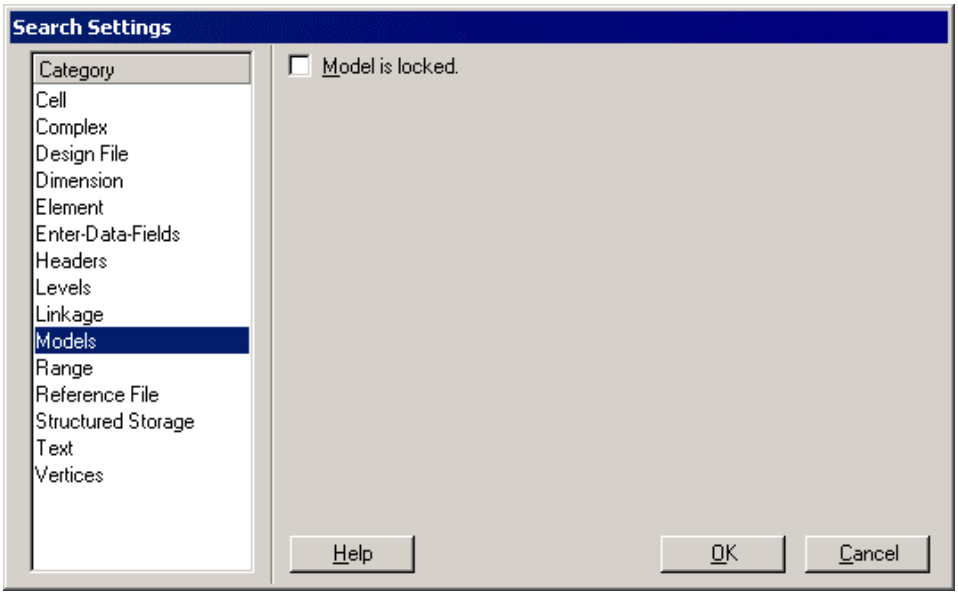
The error number reported is 764.

User linkage is not a multiple of 4 words.

The specific user linkage mentioned is not a multiple of four words in length. Note that this error is different from error 301 in that this error refers to the size of a particular attribute linkage as determined by analyzing a *specific* user linkage rather than the *entire* linkage area. The attribute area of an element may contain several individual attribute linkages. Error 301 refers to the size of the *entire* linkage area.

The error number reported is 365.

Models



"Search Settings", "Models" category.

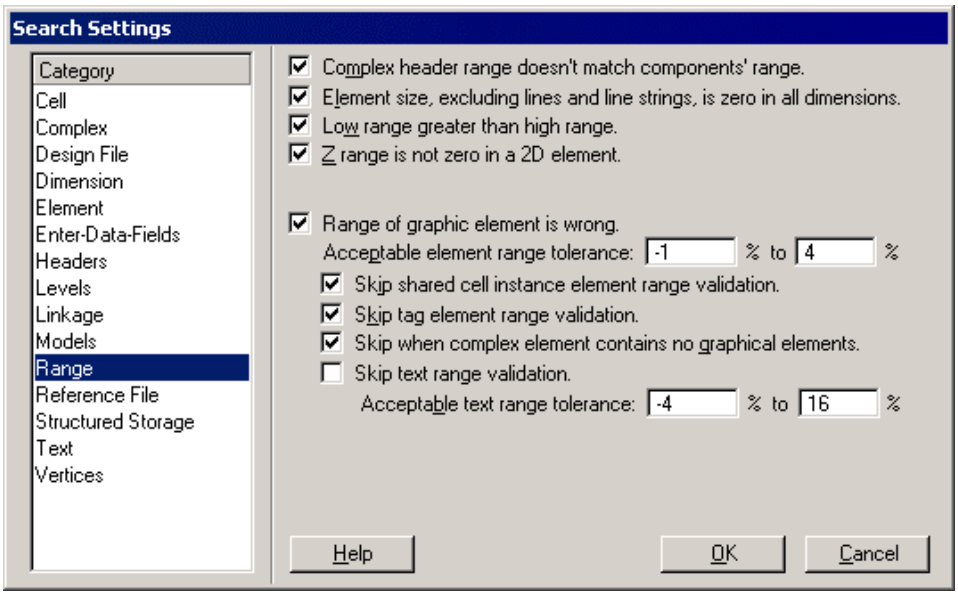
Model is locked.

The model is locked. Elements cannot be added, modified or deleted in a locked model.

The repair setting that unlocks models is *Settings* | *Change Settings* | *Repair...* | *Models* | *Unlock models*.

The error number reported is 250.

Range



"Search Settings", "Range" category.

Complex header range doesn't match components' range.

The union of the range blocks of the components of the complex element header does not match the range block of the complex header element itself. The percentage is determined by dividing the deviation from perfection as calculated by *FileFixer* (in UORs) by the size of the element (in UORs). This test is performed on the x, y, and (for 3D models) z axes. The worst (largest) of the error percentages is the one reported.

The repair setting that corrects this error is *Settings* | *Change Settings* | *Repair...* | *Range* | *Repair range of complex header elements*.

The error numbers reported are 320 and 720. Error number 320 is generated when the range is off by more than the user specified tolerance, but within the tolerance –70% and 900%.

Element size, excluding lines and line strings, is zero in all dimensions.

The size of the element in all dimensions is zero. You may not be able to see this element.

To have *FileFixer* delete such elements, add this error number (185) to the *Settings* | *Change Settings* | *Repair...* | *Element* | *Remove elements with the following error numbers* list.

The error number reported is 185.

Low range greater than high range.

The x-low value is greater than the x-high value, the y-low value is greater than the y-high value, or the z-low value is greater than the z-high value. The z-range is checked for 3D models only.

This search setting also controls whether or not *FileFixer* reports instances where the z-low or z-high range value is non-zero for an element in a 2D model.

The error number reported is 704.

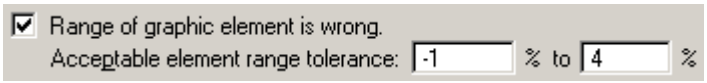
Z range is not zero in a 2D element

One or both of the z-range values is non-zero in a 2D element.

The repair setting that corrects this error is *Settings* | *Change Settings* | *Repair...* | *Range* | *Repair range of displayable elements*.

The error number reported is 169.

Range of graphic element is wrong.



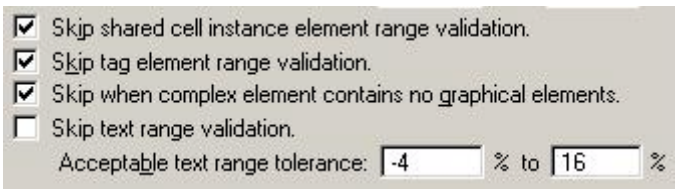
This setting tests ranges of many types of graphic elements and therefore can generate many different error codes.

“Acceptable element range of tolerance:” applies to all range tests and can be changed. The image above shows the default.

This search setting activates a number of different error numbers for different element types. There are 300-level error numbers for ranges which mildly stray from the user-specified tolerance and 700-level counterparts when the range is severely wrong.

In the process of checking ranges, *FileFixer* may encounter error 912 — part of an arc or ellipse element stored in floating point format contains an illegal value.

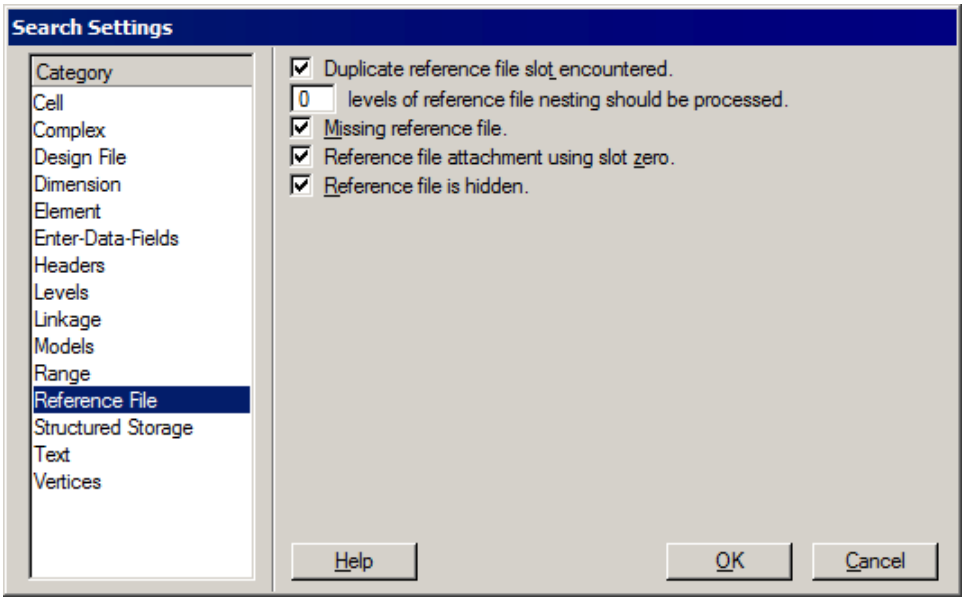
By default *FileFixer* doesn’t check for range errors on shared cell instances and tag elements. The reason for this is that to correctly calculate the range of these elements the design file has to be loaded. If there are certain types of invalid data in the design file, MicroStation may not be able to load the design file. When an MDL application tells MicroStation to load a design file and MicroStation can’t, then MicroStation will terminate all MDL applications currently running (which means *FileFixer* will be unloaded). So once you know you can load the design file, you can turn off one or both of the “Skip” search settings for these element types.



The search setting “Skip text range validation” is used if you don’t want to test for range errors on text elements. This is useful when you don’t have the font library or font resource file.

The repair setting that corrects this error is *Settings* | *Change Settings* | *Repair...* | *Range* | *Repair range of displayable elements*.

Reference File



“Search Settings”, “Reference file” category.

Duplicate reference file slot encountered

MicroStation only displays the first occurrence of a reference file with a particular slot number in a model — any other attachment using the same slot number won’t display and can’t be deleted from MicroStation’s References dialog.

The repair setting that fixes the duplicate slot number is *Settings* | *Change Settings* | *Repair...* | *Reference File* | *Set duplicate slot number to an unused slot number*.

The error number is 314.

___ levels of reference files should be processed.

This search setting allows you to select the nesting level of reference files that will be processed.

If you enter 0, no reference files are processed.

If you enter 1, the reference files *directly* attached (one level of nesting) will also be processed.

If you enter **n**, **n** levels of nested reference files will be processed.

Missing reference file.

When selected, this search setting will report deleted or moved reference files.

The error number reported is 343.

Reference file attachment using slot zero.

MicroStation assigns each reference file attachment to a slot number starting with slot number 1. This option detects a reference attachment erroneously assigned to slot zero. These attachments will display if found, but other applications like ProjectWise do not recognize an attachment assigned to slot zero.

The repair setting that fixes attachments using slot zero is *Settings* | *Change Settings* | *Repair...* | *Reference File* | *Move slot zero attachment to:* | *The first available slot* or *The slot after the highest numbered slot currently used.*

The error number is 420.

Reference file is hidden.

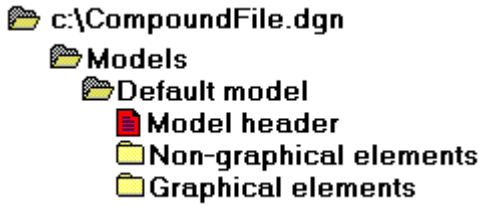
When selected, *FileFixer* will report reference file attachments which are hidden (not displayed in MicroStation's "References" dialog).

The repair setting that makes hidden reference files visible is *Settings* | *Change Settings* | *Repair...* | *Reference File* | *Display hidden reference files*.

The error number reported is 443.

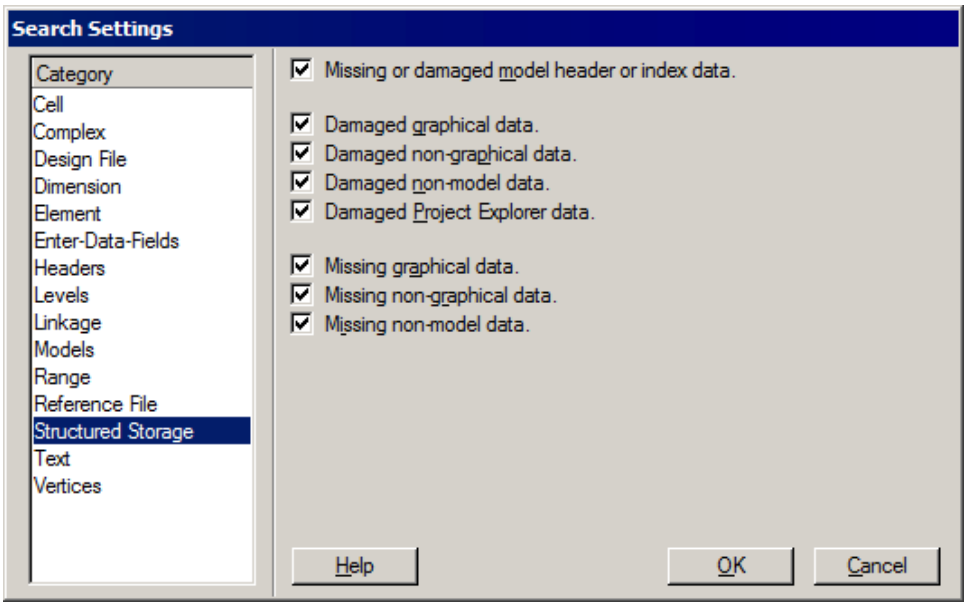
Structured Storage

Each MicroStation V8 file is actually a compound file consisting of multiple internal directories and sub-files. For example:



The above illustration represents a simplified view of *one* model ("Default") in a V8 design file. In production, V8 design files can contain *dozens* of sub-folders and sub-files.

Some of these sub-folders and sub-files, when damaged, impact users and MicroStation more severely than others. For more information, see the section "Fatal V8-specific Errors".



"Search Settings", "Structured Storage" category.

Missing or damaged model header or index data.

Each V8 model (or cell) includes a model *header* which stores the model's name, description, type and other model attributes. The model information, for all models, is also stored in the model index.

A missing or damaged model header or model index can make that model or models (and often the entire file) unusable.

Note: At this time, there is no known benefit to disabling this setting.

For more information see the “Repair” section for “Structured Storage”.

The error numbers reported are 801 (missing), 802 (damaged) and 820 (fatal).

The repair setting that corrects this error is *Settings* | *Change Settings* | *Repair...* | *Structured Storage* | *Rebuild missing or damaged model header or index data*.

Damaged graphical data.

Graphical data includes the visible geometry in your design.

Unless this setting is disabled, *FileFixer* will notify you if a model’s graphical data is damaged.

The error number reported is 802.

The repair setting that corrects this error is *Settings* | *Change Settings* | *Repair...* | *Structured Storage* | *Recover elements beyond missing or damaged graphical data*.

Damaged non-graphical data.

Non-graphical data includes reference file attachment elements and type 66 elements.

When this setting is enabled, *FileFixer* will notify you when a model’s non-graphical data is damaged.

The error number reported is 802.

The repair setting that corrects this error is *Settings* | *Change Settings* | *Repair...* | *Structured Storage* | *Recover elements beyond missing or damaged non-graphical data*.

Damaged non-model data.

Non-model data includes the design file header, level definitions, shared cell definitions, font definitions and other information that is not model-specific.

When this setting is enabled, *FileFixer* will notify you when the non-model data is damaged.

The error number reported is 802.

The Repair setting that corrects this error is *Settings* | *Change Settings* | *Repair...* | *Structured Storage* | *Recover elements beyond missing or damaged non-model data*.

Damaged Project Explorer data.

Project Explorer data can become corrupt and cause a design file to take a very long time to open. This search option determines when the Project Explorer data is damaged.

The error number reported is 450.

The Repair setting that corrects this error is *Settings* | *Change Settings* | *Repair...* | *Structured Storage* | *Repair Project Explorer data.*

Missing graphical data.

Graphical data includes the *visible* geometry (lines, circles, text, cells, etc.) in your design.

When this setting is enabled, *FileFixer* will notify you when a model's graphical data is missing.

The error number reported is 801.

The repair setting that corrects this error is *Settings* | *Change Settings* | *Repair...* | *Structured Storage* | *Recover elements beyond missing or damaged graphical data*.

Missing non-graphical data.

Non-graphical data includes reference file attachment elements and type 66 elements.

When this setting is enabled, *FileFixer* will notify you when a model's non-graphical data is missing.

The error number reported is 801.

Missing non-model data.

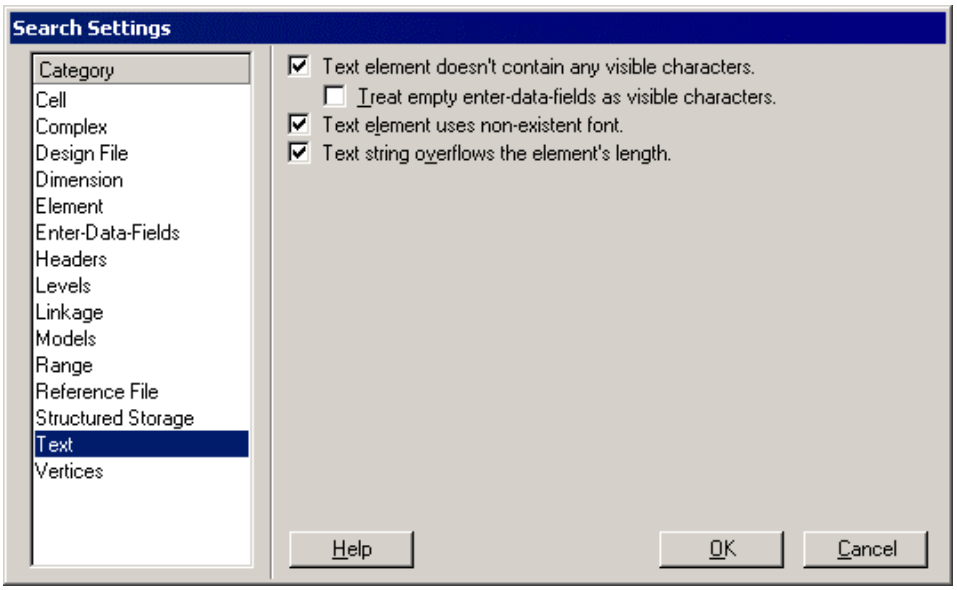
Non-model data includes the design file header, level definitions, shared cell definitions, font definitions, and other information which is not model-specific.

When this setting is enabled, *FileFixer* will notify you when the non-model data is missing.

The error number reported is 801.

The repair setting that corrects this error is *Settings* | *Change Settings* | *Repair...* | *Structured Storage* | *Recover elements beyond missing or damaged non-model data*.

Text



"Search Settings", "Text" category.

Text element doesn't contain any visible characters.

Use this setting to identify text elements which can't contain any visible characters — for example, text elements which only contain spaces.

By default *FileFixer* does not analyze text elements that contain enter-data-fields. If you turn on the "Treat empty enter-data-fields as visible characters" setting *FileFixer* will process text elements which contain enter-data-fields.

The error number reported is 146.

Text element uses non-existent font.

The font library MicroStation is currently using does not contain the font used by this text element.

The error number reported is 144.

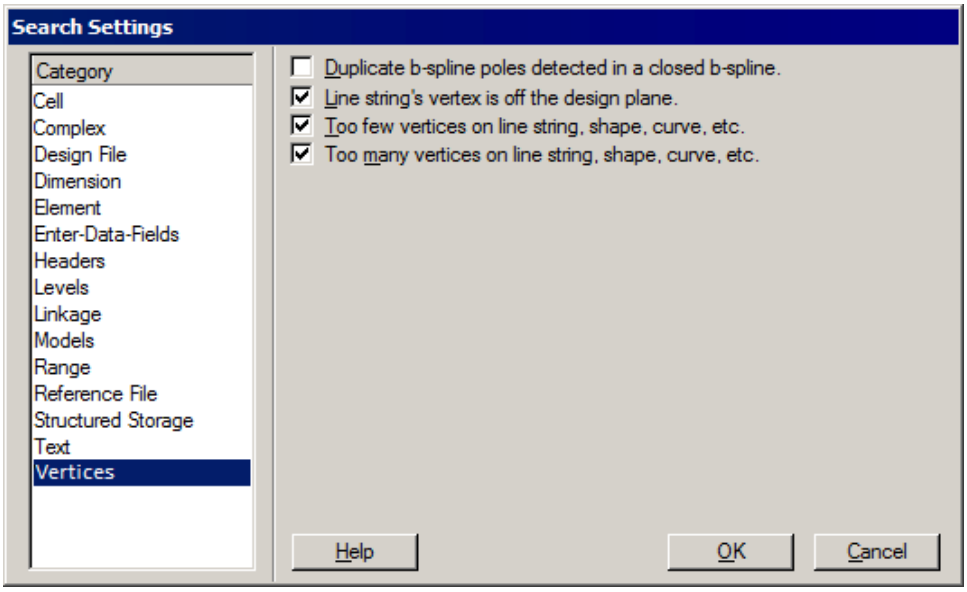
Text string overflows the element's length.

The number of characters is incorrect. It says there are more characters than actually exist.

The repair setting that corrects this error is *Settings* | *Change Settings* | *Repair...* | *Text* | *Correct the number of characters field in text element.*

The error number is 560.

Vertices



"Search Settings", "Vertices" category.

Duplicate b-spline poles detected in a closed b-spline.

A b-spline pole element (type 24) normally doesn't have duplicate first and/or last poles, and its first and last poles normally don't match.

To avoid problems, for example to retain backward compatibility with MicroStation SE, the solution is to change the closed b-spline to an open b-spline.

The repair setting that corrects this problem is *Settings* | *Change Settings* | *Repair...* | *B-spline* | *Open the closed b-spline that has duplicate b-spline poles*.

The error number reported is 905.

Line string's vertex is off the design plane.

One of the line string's vertices is off the design plane.

The repair setting that corrects this problem is *Settings* | *Change Settings* | *Repair...* | *Vertices* | *Fix line string's vertex which is off the edge of the design plane by*. The repair feature either allows you to adjust the corrupted vertex or delete the element. To adjust the corrupted vertex the vertices on both sides of it must be on the design plane (otherwise the vertex cannot be adjusted).

The error number reported is 373.

Too few vertices on line string, shape, curve, etc.

Error number 788 is reported when a type 21 (b-spline pole) element has less than two poles.

Error number 789 is reported when elements with vertices, points, etc. do not have a minimum number of these.

The repair setting that detects error 789 is *Settings* | *Change Settings* | *Repair...* | *Element* | *Remove elements with too few vertices*.

Too many vertices on line string, shape, curve, etc.

Error number 787 is reported when elements with vertices have a “number of vertices” value greater than the maximum allowed of 5000.

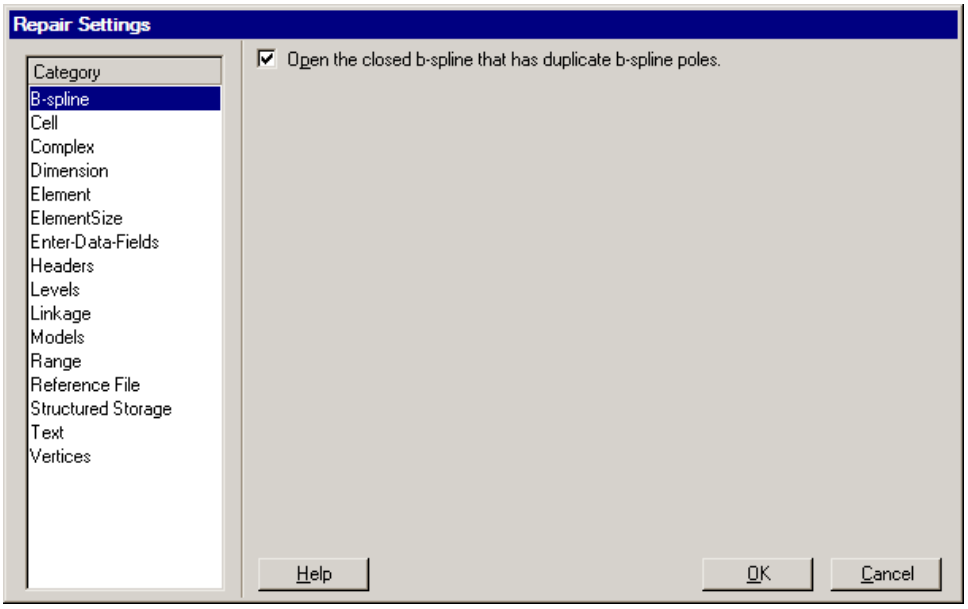
The repair setting that fixes error 787 is Settings | Change Settings | Repair... | Vertices | Set number of vertices to maximum allowed.

Chapter 9 — Repair Settings

Factory defaults

Unless indicated otherwise, the settings illustrated in the following screen images reflect *FileFixer's* factory default settings.

B-spline



"Repair Settings", "B-spline" category.

Open the closed b-spline that has duplicate b-spline poles

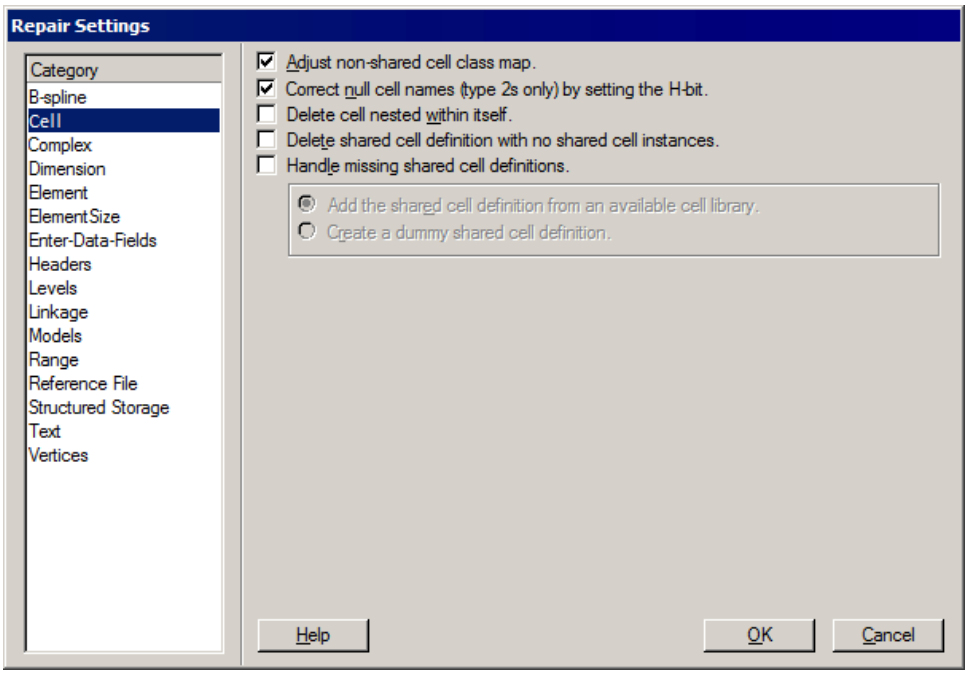
A b-spline pole element (type 21) normally doesn't have duplicated first and/or last poles, and its first and last poles normally don't match.

To avoid problems, for example to retain backward compatibility with MicroStation SE, change the closed b-spline to an open b-spline.

The search setting that detects this error is *Settings* | *Change Settings* | *Search...* | *Vertices* | *Duplicate b-spline poles detected in a closed b-spline*.

The error number reported is 905.

Cell



"Repair Settings", "Cell" category.

Adjust non-shared cell class map.

This repair setting corrects the list of classes (which is stored in the cell header element) used by elements in the cell.

The search setting that detects this error is *Settings* | *Change Settings* | *Search...* | *Cell* | *Non-shared cell class map is incorrect*.

The error number that is reported is 717.

Correct null cell names (type 2s only) by setting the H-bit.

In this context, "null" means *empty* — no cell name. A type 2 cell in a cell library is not required to have a name, but nameless type 2 cells should not exist in a cell library unless their H-bit is set to indicate they are "orphan cells".

FileFixer will ensure nameless type 2 cells have their H-bit set.

The repair setting is "Correct null cell names (type 2s only) by setting the H-bit." This setting is ON by default.

The error number reported is 570.

Delete cell nested within itself.

Use this setting to delete any cell nested within itself. All duplicate nested cells will be deleted.

The search setting that detects this error is *Settings* | *Change Settings* | *Search...* | *Cell* | *Cell is nested within itself.*

The error number that is reported when the nested cell is in a design file is 383.

This setting is OFF by default.

Delete shared cell definition with no shared cell instances.

Delete the shared cell definition that has no shared cell instance. This happens quite often when importing DWG files to MicroStation.

The search setting that detects this error is *Settings* | *Change Settings* | *Search...* | *Cell* | *Shared cell definition with no shared cell instance*.

The error number reported is 303.

Handle missing shared cell definitions.

When a shared cell instance is found with no corresponding shared cell definition, then use this repair setting to fix it in one of two ways.

Add the shared cell definition from an available cell library. When this option is chosen *FileFixer* creates a shared cell definition using a cell with the same name found in one of the cell libraries searched in the active MicroStation workspace. If a matching cell name is not found *FileFixer* will report that it could not be found in the report.

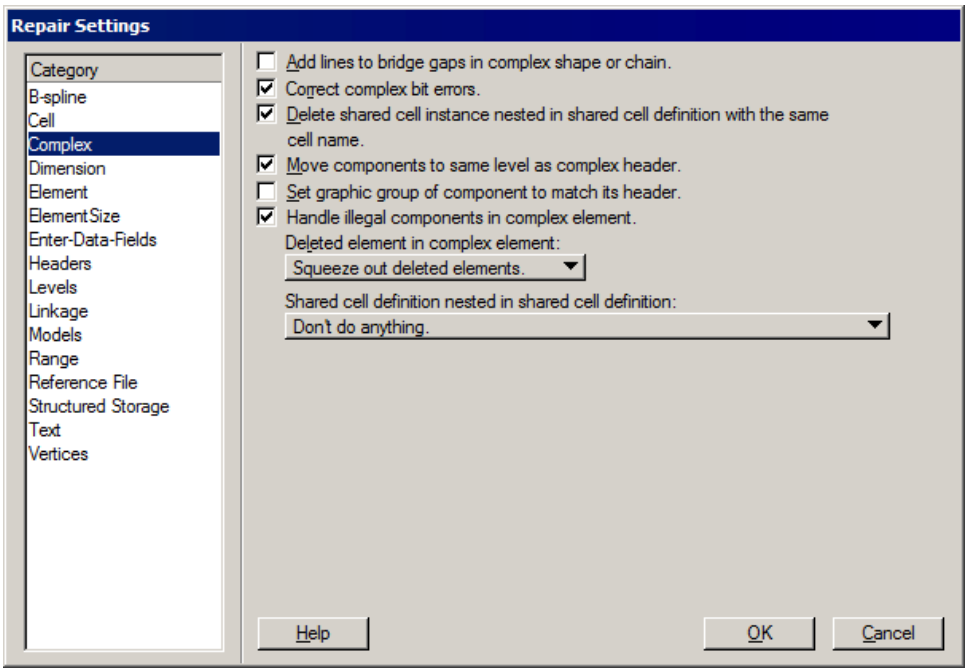
Note: *FileFixer* uses its own seed file during repair, therefore it will not know of a cell library that is attached to the original file being fixed. For this reason it is important that the cell libraries containing the cells you want use are found in one of the directories MicroStation searches in the active workspace.

Create a dummy shared cell definition. When this option is chosen *FileFixer* creates a dummy shared cell definition with a single component text element that displays the shared cell’s name.

The search setting that detects this error is *Settings* | *Change Settings* | *Search...* | *Cell* | *Shared cell instance has no shared cell definition*.

The error number reported is 768.

Complex



"Repair Settings", "Complex" category.

Add line to bridge gap in 2D complex shape or chain.

When creating a 2D complex shape or chain from multiple linear elements that have gaps between the endpoints, MicroStation will not actually create the connecting lines, but MicroStation will *display* a line between those endpoints. If these complex elements are later dropped, gaps will exist between the component elements.

The search setting that detects this error is "*Settings* | *Change Settings* | *Search...* | *Complex* | *Gap exists in 2D complex shape or chain*. This repair option is off by default.

The error number reported is 151.

Correct complex bit errors.

FileFixer clears the complex bit for an element that is not a component of a complex element but does have the complex bit set. The error number reported for this error is 702.

FileFixer sets the complex bit for an element that is a component of a complex element but does not have its complex bit set. The error number reported for this error is 703.

The search settings that detect these errors are *Settings* | *Change Settings* | *Search...* | *Complex* | *Complex bit is on for a non-component element* and *Settings* | *Change Settings* | *Search...* | *Complex* | *Complex bit is off for a component element.*

Delete shared cell instance nested in a shared cell definition with the same cell name.

Use this repair setting to delete the shared cell instance nested in a shared cell definition with the same name.

The search setting that detects this error is *Settings* | *Change Settings* | *Search...* | *Complex* | *Shared cell instance nested in a shared cell definition with the same name.*

The error number reported is 733.

Move components to same level as complex header.

Use this repair setting to move the component element to the same level as the complex header.

The search setting that detects this error is *Settings* | *Change Settings* | *Search...* | *Complex* | *Component on different level than its header*.

The error number reported is 336.

Set graphic group of component to match its header.

FileFixer changes the graphic group number of the component element to match the header.

This error is not reported for component elements that are part of regular cells (type 2), shared cell definitions (type 34) or raster headers (type 87 or type 90).

This error is not reported for components of non-graphical complex headers, such as level table entries, etc.

The search setting that detects this error is *Settings* | *Change Settings* | *Search...* | *Complex* | *Component graphic group doesn't match header*.

The error number reported is 341.

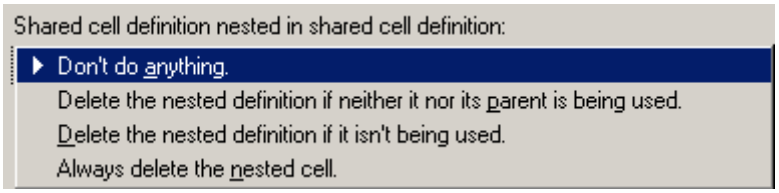
Handle illegal components in complex element.

For example, an ellipse should not be a component in a text node. When *FileFixer* encounters an illegal component, the complex element is normally broken up into its component parts. (Traditional jargon for this is to "drop complex status" or "drop status".)

There are two special cases where other settings are available:

When *deleted* components are found in a complex header, *FileFixer* allows you to either break up the complex element into its component parts (traditional jargon is to "drop status on the complex header") or remove ("squeeze out") the deleted component elements.

The following specialized repair settings are available for nested shared cell definitions (type 34):

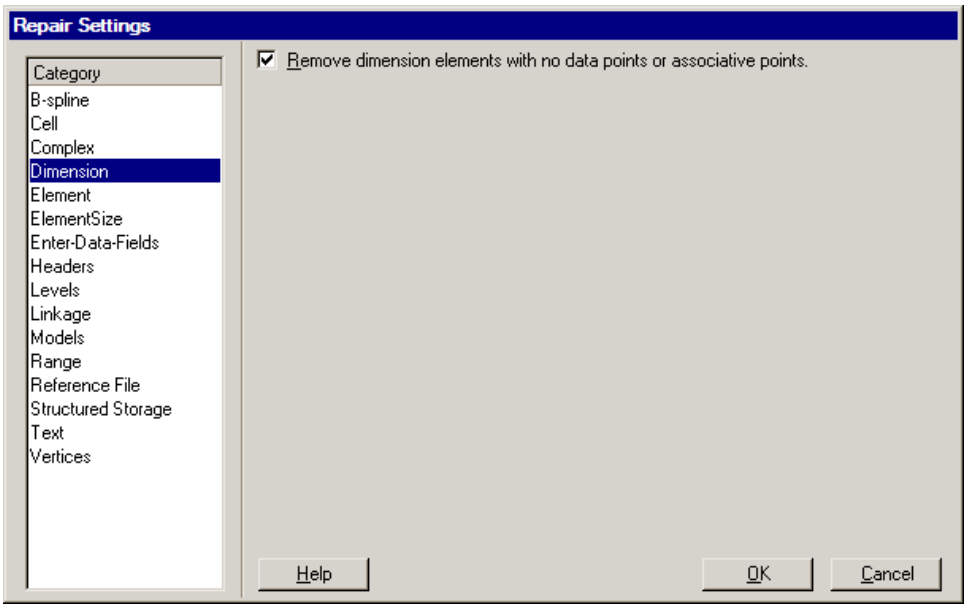

Shared cell definition nested in shared cell definition:
Don't do anything.
Delete the nested definition if neither it nor its parent is being used.
Delete the nested definition if it isn't being used.
Always delete the nested cell.

You won't want to automatically delete the nested shared cell if it is referenced by a shared cell instance (type 35) or if it is used as a pattern. In these cases you can choose to convert the outer shared cell definition to a cell. The default handling with nested shared cells is "Don't do anything".

The search setting that detects this error is *Settings* | *Change Settings* | *Search...* | *Complex* | *Complex element contains disallowed element type*.

The error number reported is 721. This search setting also controls error numbers 121 and 323, but the only error *repaired* is 721.

Dimension



"Repair Settings", "Dimension" category.

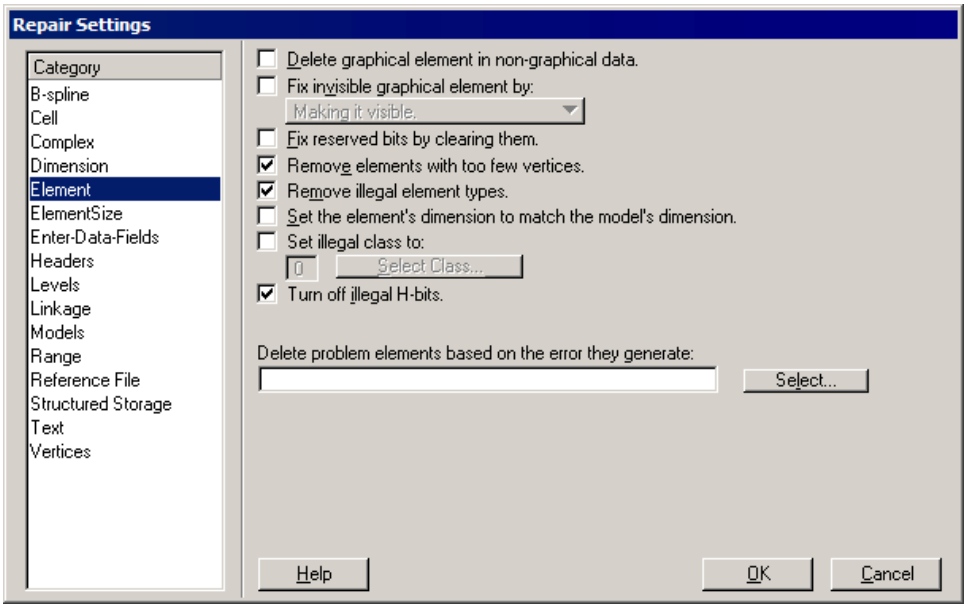
Remove dimension element with no data points or associative points.

Remove the dimension element that doesn't have at least one data point or associative point. An associative point is a point that ties this dimension element to another element in the design file.

The search setting that detects this error is *Settings* | *Change Settings* | *Search...* | *Dimension* | *Dimension element with no data points or associative points*.

The error number reported is 722.

Element



"Repair Settings", "Element" category.

Delete graphical element in non-graphical data.

Graphical elements in the non-graphical data area can cause MicroStation V8i to lock up or abort. This data area is for reference file attachments, application data elements, etc.

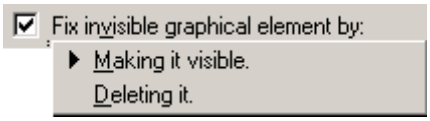
The search setting that detects this error is *Settings* | *Change Settings* | *Search* | *Element* | *Graphical element in non-graphical data*.

The error number reported is 819.

Fix invisible graphical element by:

MicroStation V8 has the ability to make an element "invisible". It will not be displayed on MicroStation's GUI or be plotted. This is a new MicroStation feature so these elements are not really corrupted, which is why this setting is OFF by default.

If you use Cimmetry's AutoVue application, which does display these elements, then the AutoVue GUI can be a little confusing. This is the reason for this setting.



With this repair setting *FileFixer* either makes visible or deletes “invisible” graphical elements. *FileFixer* will not make visible or delete pattern control elements of tag elements.

The search setting that detects this error is *Settings* | *Change Settings* | *Search...* | *Element* | *Graphical element is invisible.*

The error number reported is 240.

Fix reserved bits by clearing them?

This repair setting *FileFixer* turns off the six reserved bits in element headers.

The search setting that detects this error is *Settings* | *Change Settings* | *Search...* | *Element* | *Reserved bits are set in element header.*

The error number reported is 103.

Remove elements with too few vertices.

With this repair setting *FileFixer* will remove any element with too few vertices.

Error number 788 is reported when a b-spline pole has less than two poles.

Error number 789 is reported when elements with vertices, points, etc., do not have the minimum number.

The search setting that detects this error is *Settings* | *Change Settings* | *Search...* | *Vertices* | *Too few vertices on line string, shape, curve, etc.*

Remove illegal element types.

FileFixer will delete illegal element types detected, such as type 0, 20 and 127. The error number reported is 380.

Error number 380 is controlled by the search setting *Settings* | *Change Settings* | *Search...* | *Type/Level* | *Illegal element type (0, 1, 20, 127, etc.).*

Note: Also see description for error 302 — Type 1 (V7 cell library header) elements are not allowed in design files.

Set the element’s dimension to match the model’s dimension.

Set the element’s dimension to the model’s dimension.

The search setting that detects this error is *Settings* | *Change Settings* | *Search ...* | *Element* | *Element’s and model’s dimensions are different*.

The error number reported is 437.

Set illegal class to:

Set the illegal class (greater than 6) to the class you have specified.

The search setting that detects this error is *Settings* | *Change Settings* | *Search...* | *Element* | *Element class is illegal (must be less than 7).*

The error number reported is 333.

Turn off illegal H-bits.

The H-Bit means different things for different element types. The only elements whose H-Bit should be set are cells (type 2), lines (type 3), shapes (type 6), complex shapes (type 14), ellipses (type 15), and element types 21 through 28. The reported element was not one of these types, yet its H-Bit was set.

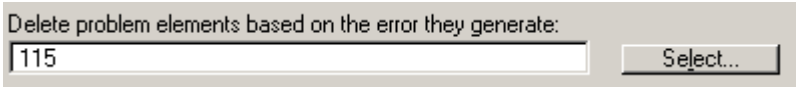
The search setting that detects this error is *Settings* | *Change Settings* | *Search...* | *Element* | *H-bit is set when it shouldn't be.*

The error number reported is 304.

Delete problem elements based on the error they generate.

This setting is provided for user convenience. If you decide to delete one or more problem elements based on the error number they generate, use this setting.

Example:



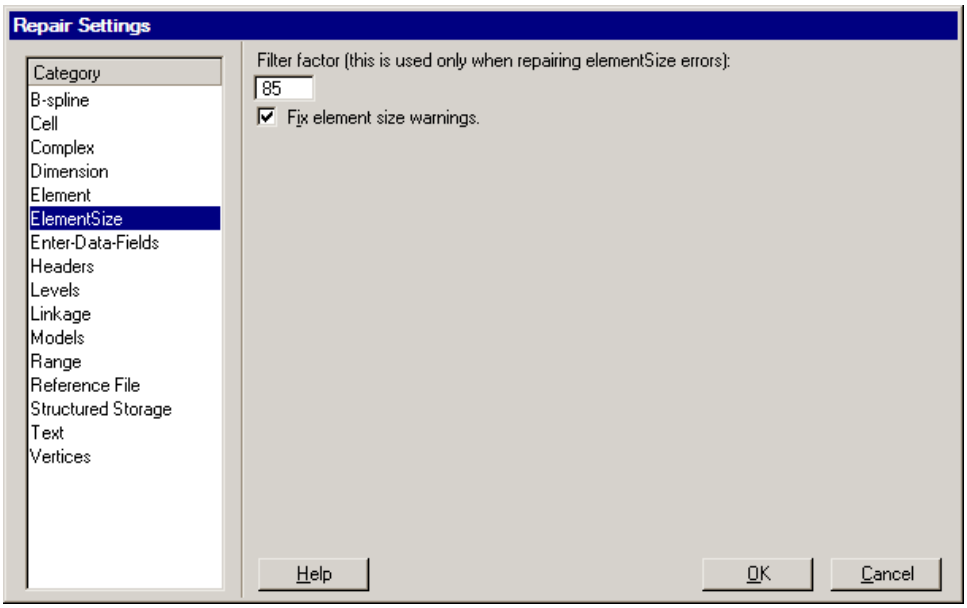
You can enter a series of error numbers separated by commas.

The <Select> button allows you to select from a list of error numbers and their descriptions.

FileFixer will delete all elements with matching error numbers.

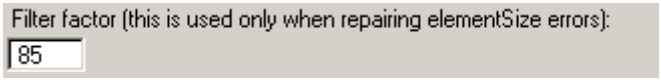
Note: If the element that generated the error is a complex header element, *FileFixer* will break up the complex element into its individual components and delete just the complex header. You can use *Problem Element Viewer* to locate, view and delete all components.

ElementSize



“Repair Settings”, “ElementSize” category.

Filter factor



“Filter factor” is used only when repairing elementSize errors.

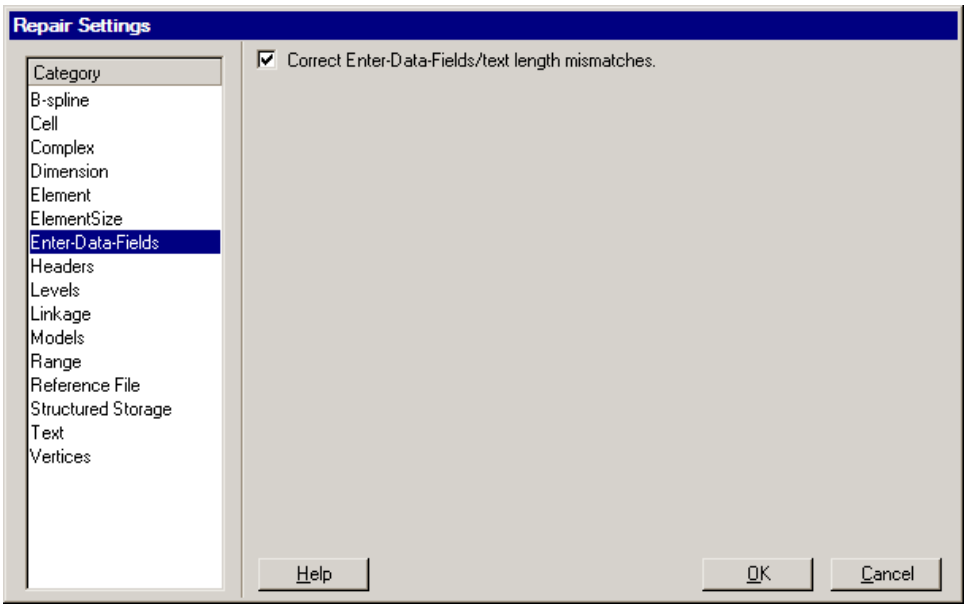
The default “Filter factor” usually works well, but trying different values can improve results in some cases.

The “Filter factor” determines how selective *FileFixer* is in determining when it has encountered a valid element chain. If the “Filter factor” is too low, *FileFixer* may think it has encountered a valid element chain when it has not. If the “Filter factor” is too high, *FileFixer* might ignore a valid element chain.

Fix element size warnings.

This repair setting directs *FileFixer* to fix error number 364 — the element’s attribute linkage extends beyond the end of the element. This means the element’s linkage overlaps the *next* element.

Enter-Data-Fields



"Repair Settings", "Enter-Data-Fields" category.

Correct EDF/text length mismatches.

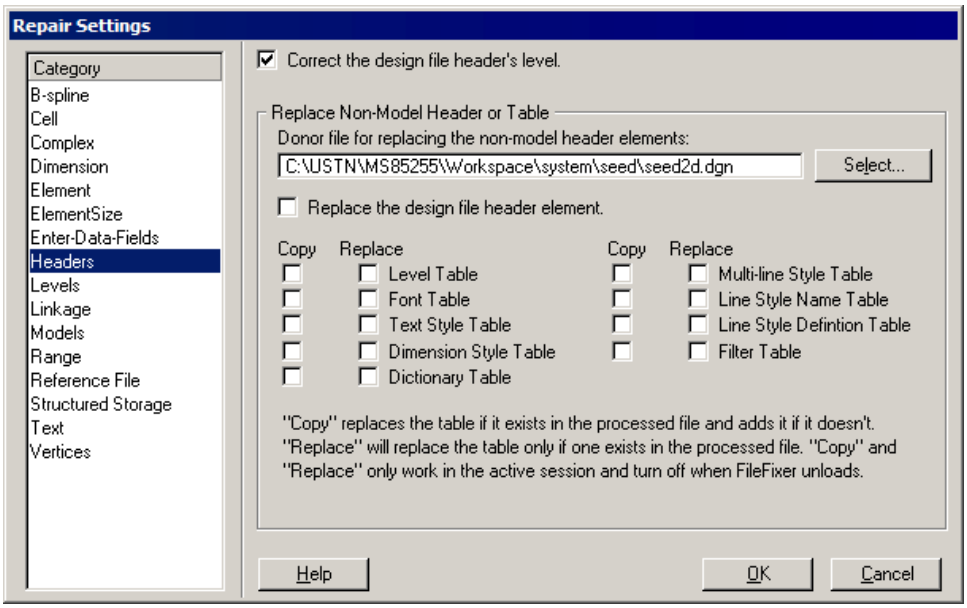
FileFixer deletes Enter-Data-Fields (EDFs) when there are more than 20 and when they exceed the length of the element.

FileFixer extends the text string when there are 20 or fewer EDFs and they don't exceed the element's length so that the largest Enter-Data-Field completes within the text string.

The search setting that detects the 191 error is *Settings* | *Change Settings* | *Search...* | *EDF* | *Text element has more EDFs than characters.*

The error number reported is 191.

Headers



"Repair Settings", "Headers" category.

Correct the design file header's level

FileFixer set the design file header's level to 8.

When the design file header's level number is not 8 then the "Unable to open design file. Please contact your local site administrator or technical support provider for further assistance." message is displayed by MicroStation V8.

The search setting that detects this error is *Settings* | *Change Settings* | *Search...* | *Headers* | *Design file header is on the wrong level*.

The error number reported is 911.

Replace Non-Model Header or Table

There are a number of header elements and tables in the non-model section of the V8 design file. This section gives you the ability to replace them when you are having a problem with one or more of them.

Replace the design file header element.

The design file header (type 9) element is the first element in a MicroStation design file. It stores a variety of vital information.

Design file header corruption can manifest itself in a number of ways:

- Unable to open the design file in MicroStation.
- FIT command behaves oddly.

- After loading the file, MicroStation aborts.
- Text elements do not display correctly.
- Other odd, inexplicable behavior.

FileFixer can replace a corrupt design file header with a healthy one.

The error number is 21.

Note: *FileFixer* will never attempt to replace a design file header unless you explicitly activate the option *Settings | Repair Settings | Replace the design file header element* in the active session. Once you close *FileFixer* this option is turned off. Accordingly, error number 21 will never be reported by *FileFixer*'s Search for Problems command. There is no Search setting which enables or disables error number 21.

Copy or Replace Tables

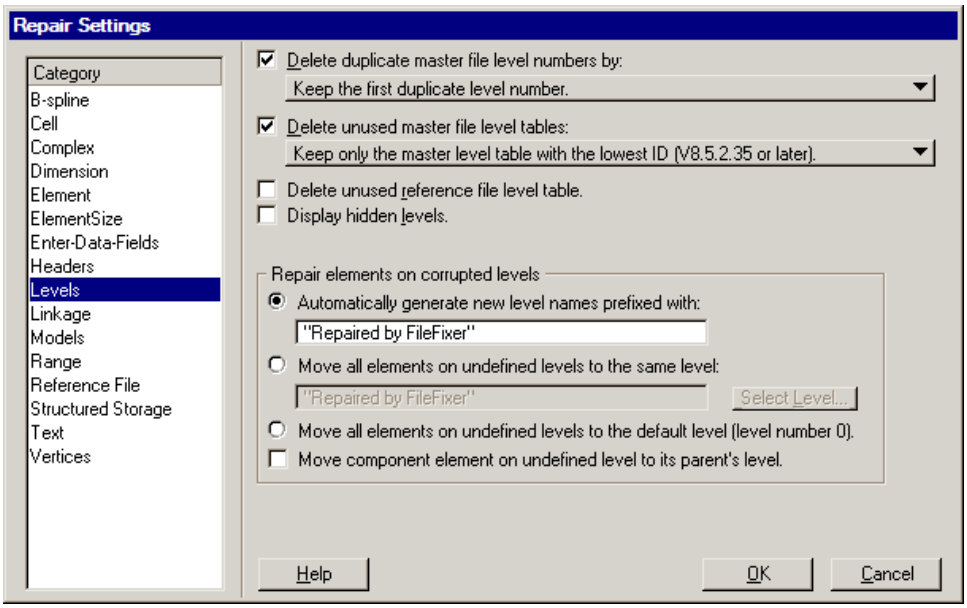
You have the ability to copy or replace the level table, font table, text style table, dimension style table, dictionary table, multi-line style table, line style name table, line style definition table or the filter table.

Copy and replace work slightly differently. Replace will only replace the table if it already exist. Copy will replace the table if it already exists or just copy it in if it isn't already in the design file.

The error number is 22.

Note: *FileFixer* will never attempt to replace a non-model table unless you explicitly activate the copy or replace option for that non-model table in the active session. Once you close *FileFixer* all "Copy" and "Replace" settings you have made are turned off. Accordingly, error number 22 will never be reported by *FileFixer*'s Search for Problems command. There is no Search setting which enables or disables error number 22.

Levels



"Repair Settings", "Levels" category.

Delete duplicate master file level numbers by:

The level *number* displayed by MicroStation (in the Level Manager dialog) is user-editable and, therefore, not guaranteed to be unique. *Internally* (not viewable by users) MicroStation assigns a *unique* level "code" to each defined level.

By default, *FileFixer* keeps the first duplicate level table entry associated with a master file level *number*. You can optionally keep the last level table entry for a duplicate master file level *number*.

Note: *FileFixer* always retains the "Default" level definition regardless of the repair option selected here. This is a MicroStation requirement.

The search setting that detects this error is *Settings* | *Change Settings* | *Search...* | *Levels* | *Multiple level numbers*.

The error number is 429.Delete unused master file level tables

FileFixer allows you to keep the master file level table with the lowest ID (typically the original) or to keep the first master file level table.

MicroStation V8.5.2.35 and later use the master file level table with the lowest ID (typically the original). Earlier versions of MicroStation V8 use the first master file level table found in the design file.

The search setting that detects this error is *Settings* | *Change Settings* | *Search...* | *Levels* | *Multiple master file level tables*.

The error number reported is 630.

Delete unused reference file level tables.

Select this setting if you would like *FileFixer* to delete unused reference file level tables. The search setting that detects this error is *Settings* | *Change Settings* | *Search...* | *Levels* | | *Unused reference file level tables*.

The error number is 267.

Display hidden levels

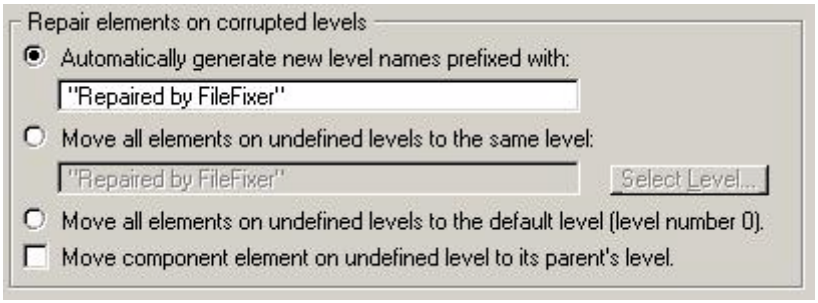
FileFixer makes hidden levels displayable in MicroStation's Level Manager.

The search setting that detects hidden levels is *Settings* | *Change Settings* | *Search...* | *Levels* | *Level is hidden*.

The error number reported is 444.

Repair elements on corrupted levels

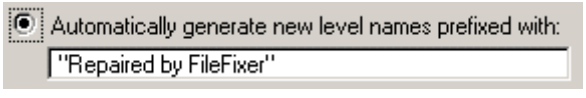
When error 415 appears in your *FileFixer* report, you can select from the following level renaming strategies — whichever is best suited for your project:


Repair elements on corrupted levels
Automatically generate new level names prefixed with:
"Repaired by FileFixer"
Move all elements on undefined levels to the same level:
"Repaired by FileFixer"
Select Level...
Move all elements on undefined levels to the default level (level number 0).
Move component element on undefined level to its parent's level.

When an element is found to contain a level that is not present in the level table, *FileFixer* will change the element's level to a new level or a user-selected level.

Automatically generate new level names prefixed with

FileFixer can automatically generate a new level name for each level missing from the level table. Using the following setting you can specify the prefix for the automatically generated levels:



The default prefix is "Repaired by *FileFixer*". The level number used for the new level will be appended onto the end of the level name. The level numbers used will begin with

4000000. The level name for the first level added using the default prefix will be "Repaired by *FileFixer* 4000000".

Note: the level prefix you specify is used to create level names, so it cannot contain the following illegal characters: `<>\/"?*|,='&.`

Move all elements on undefined levels to the same level

FileFixer can also move all elements on undefined levels to the same level.

You can either use an already defined level name or specify a new level name. If the level name doesn't already exist in the level table then *FileFixer* will use the level name entered as a prefix and append the level number (starting at 4000000) onto the end of the prefix.

Move all elements on undefined levels to the default level (level number 0).

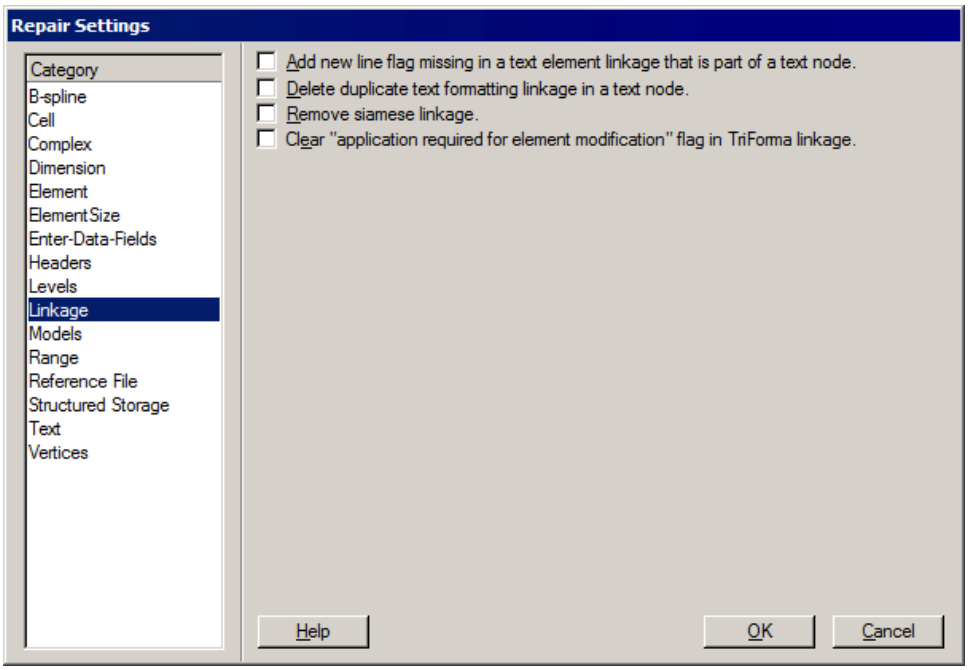
FileFixer can also move all elements to the default level. Normally, the default level will be named "Default", but not always. The default level's displayed level number will always be zero.

Move component element on undefined level to its parent's level.

FileFixer will set the corrupted component level to the same level as its complex header. This will only be done if the complex header's level isn't corrupted too. If it is then one of the other settings ("Automatically generate new level names prefixed with", "Move all elements on undefined levels to the same level" or "Move all elements on undefined levels to the default level") will be performed.

The error number reported is 415.

Linkage



"Repair Settings", "Linkage" category

Add new line flag missing in a text element linkage that is part of a text node.

Normally there is a line feed between text strings in a text node. When the component text element of a text node doesn't have its new line flag set, editing the text node will unexpectedly strip the line feeds.

When this repair option is turned on *FileFixer* will set the line feed flag in the text formatting linkage of a text node's component text elements.

The search setting that detects this error is *Settings* | *Change Settings* | *Search...* | *Linkage* | *New line flag is missing from a text element linkage in a text node.*

This repair setting is off by default.

The error number reported is 544.

Delete duplicate text formatting linkage in a text node.

Normally line feeds are maintained between text strings when editing a text node. When more than one text formatting linkage exists in each of the text elements in a text node, and the line feed flag is turned off in one or more of those linkages, editing the text node will unexpectedly strip the line feeds from the text node.

FileFixer will delete the duplicate text formatting linkages and set the line feed flag in the one remaining text formatting linkage.

The search setting that detects this error is *Settings* | *Change Settings* | *Search...* | *Linkage* | *Duplicate text formatting linkage in a text node*.

This repair setting is off by default.

The error number reported is 545.

Remove siamese linkage.

A siamese linkage occurs when a linkage overflows into the next element. This can cause many problems, such as: elements that cannot be moved without MicroStation aborting or a file that cannot be plotted.

FileFixer will delete the siamese linkage.

The search setting that detects this error is *Settings* | *Change Settings* | *Search...* | *Linkage* | *Siamese attribute linkage (linkage overlaying the next element)*.

This repair setting is off by default.

The error number reported is 764.

Clear “application required for element modification” flag in TriForma linkage.

Elements, created by TriForma, can be set to prevent MicroStation, when run without TriForma loaded, from allowing a user to modify them. This is done by setting the “application required for element modification” flag in the TriForma linkage.

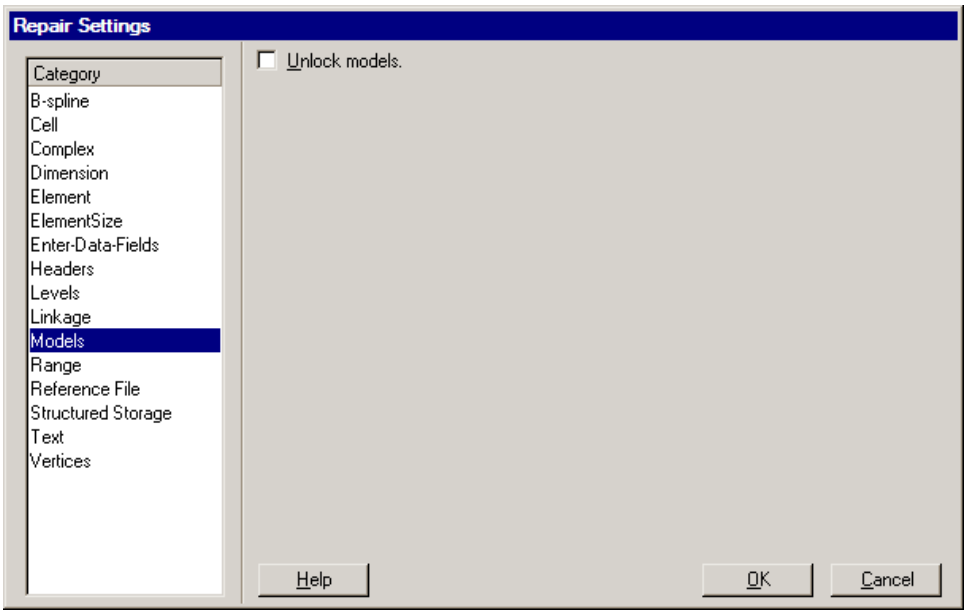
Clearing this flag with *FileFixer* allows the element to be modified without TriForma present.

This repair setting is off by default.

The search setting that detects this error is *Settings* | *Change Settings* | *Search...* | *Linkage* | *“Application required for element modification” flag in a TriForma linkage*.

The error number reported is 235.

Models



“Repair Settings”, “Models” category.

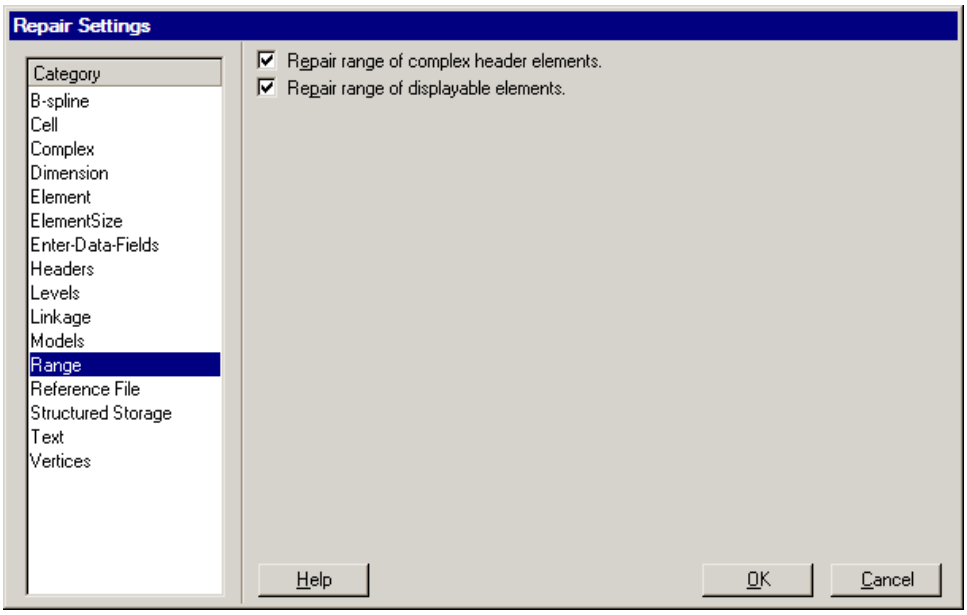
Unlock models.

Unlock a locked model. After unlocking the model you will be able to add, modify and delete elements in that model.

The search setting that detects locked models is *Settings* | *Change Settings* | *Search...* | *Models* | *Model is locked*.

The error number reported is 250.

Range



"Repair Settings", "Range" category.

Repair range of complex header elements.

Sets the range block of the complex header to be the union of the range blocks of the component elements.

The search setting that detects this error is *Settings* | *Change Settings* | *Search...* | *Range* | *Complex header range doesn't match components' range*.

The error numbers reported are 320 and 720. Error number 320 is generated when the range is off by more than the user specified tolerance, but within the tolerance –70% and 900%.

Repair range of displayable elements.

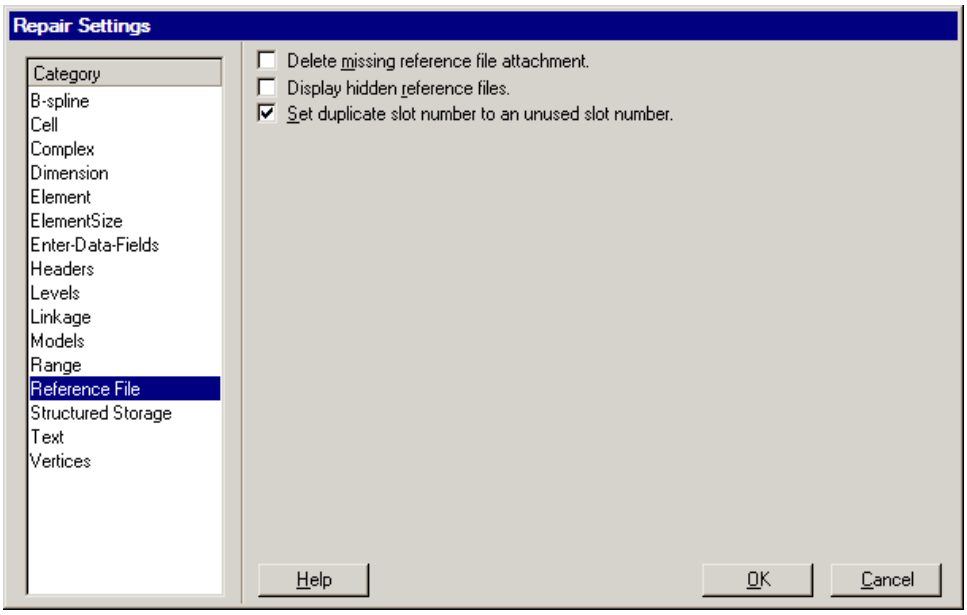
Use this repair setting to correct the range of displayable elements.

This repair setting also controls whether or not *FileFixer* repairs instances where the z-low or z-high range value is non-zero for an element in a 2D model.

The search setting that detects this error is *Settings* | *Change Settings* | *Search...* | *Range* | *Range of graphic element is wrong*.

The error numbers are 119, 169, 305, 307, 309, 312, 324, 325, 619, 704, 705, 707, 709, 712, 724, 725, and 912. Please refer to the search setting *Settings* | *Change Settings* | *Search...* | *Range* | *Range of graphic element is wrong* for more information on each of these error numbers.

Reference File



“Repair Settings”, “Reference File” category.

Delete missing reference file attachment.

Select this setting if you would like *FileFixer* to delete reference file attachments when the reference file cannot be found.

The error number is 343.

Display hidden reference files.

Select this setting if you would like *FileFixer* to make hidden reference file attachments displayable in MicroStation.

The search setting that detects this error is *Settings* | *Change Settings* | *Search...* | *Reference File* | *Reference file is hidden*.

The error number is 443.

Set duplicate slot number to an unused slot number

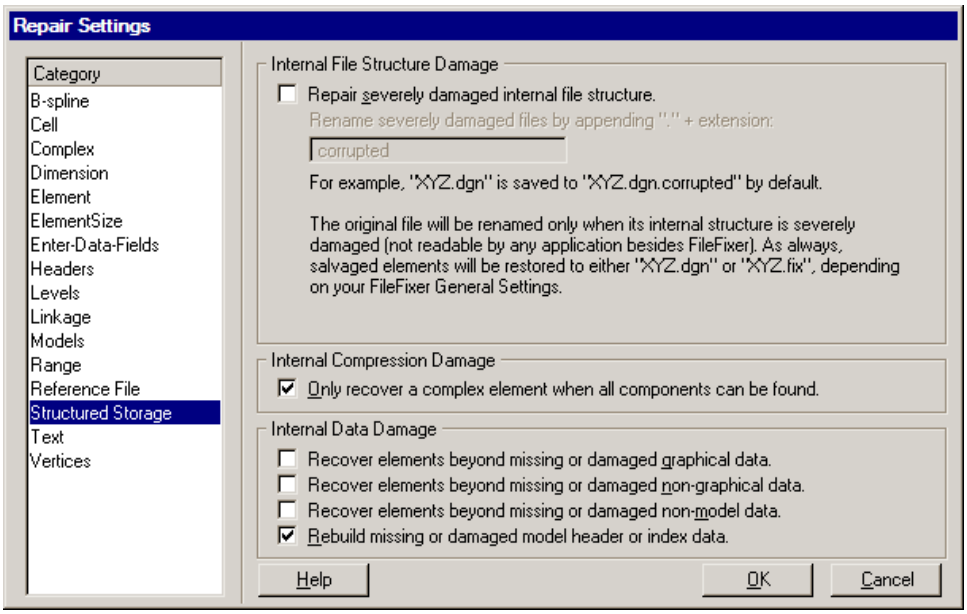
Use this option to set the duplicate slot number to one greater than the highest slot number currently being used by this model’s reference files.

The search setting that detects this error is *Settings* | *Change Settings* | *Search...* | *Reference File* | *Duplicate reference file slot encountered*.

The error number is 314.

Structured Storage

Each MicroStation V8 file is actually a compound file consisting of multiple internal directories and sub-files. This data organization scheme is also used in Microsoft Word documents and Excel spreadsheets.


Repair Settings
Category
B-spline
Cell
Complex
Dimension
Element
ElementSize
Enter-Data-Fields
Headers
Levels
Linkage
Models
Range
Reference File
Structured Storage
Text
Vertices
Internal File Structure Damage
Repair severely damaged internal file structure.
Rename severely damaged files by appending "." + extension:
corrupted
For example, "XYZ.dgn" is saved to "XYZ.dgn.corrupted" by default.
The original file will be renamed only when its internal structure is severely damaged (not readable by any application besides FileFixer). As always, salvaged elements will be restored to either "XYZ.dgn" or "XYZ.fix", depending on your FileFixer General Settings.
Internal Compression Damage
Only recover a complex element when all components can be found.
Internal Data Damage
Recover elements beyond missing or damaged graphical data.
Recover elements beyond missing or damaged non-graphical data.
Recover elements beyond missing or damaged non-model data.
Rebuild missing or damaged model header or index data.
Help
OK
Cancel

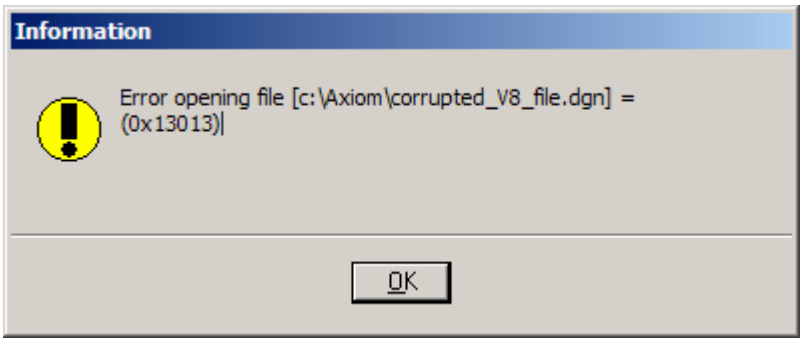
"Repair Settings", "Structured Storage" Category

Internal File Structure Damage

This section of the "Structured Storage" repair settings, gives you an option for repairing files that have severely damaged internal file structure. It also gives you an option for naming the repaired file. The following sections cover these options in detail.

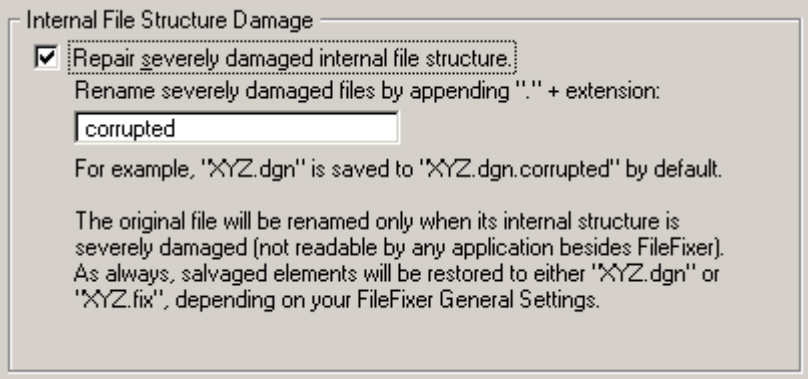
Repair severely damaged internal file structure

When the internal file structure is corrupted the directories and sub-files cannot be accessed. When you try to open the file with MicroStation you will see an error such as the following:



Using this repair setting *FileFixer* reads through the corrupted design file to identify the start of all the sub-files. *FileFixer* determines which sub-files are readable (totally intact). Healthy sub-files are used to create a clean (healthy, empty) design file. Inaccessible sub-files are not copied to the new design file.

FileFixer renames severely damaged files by adding a "." followed by a user-definable extension to the end of the file name. The extension is "corrupted" by default.


Internal File Structure Damage
Repair severely damaged internal file structure.
Rename severely damaged files by appending "." + extension:
corrupted
For example, "XYZ.dgn" is saved to "XYZ.dgn.corrupted" by default.
The original file will be renamed only when its internal structure is severely damaged (not readable by any application besides FileFixer). As always, salvaged elements will be restored to either "XYZ.dgn" or "XYZ.fix", depending on your FileFixer General Settings.

After all the available directories and sub-files have been copied into a clean V8 design file, the new file is given the original design file's name. Then *FileFixer* continues processing this new design file.

Original file:

axiom.dgn

By default, original file is saved to:

axiom.dgn.corrupted

FileFixer then reconstructs axiom.dgn and repairs it.

The error number reported is 899.

Internal Compression Damage

This section of the "Structured Storage" repair settings contains one option for recovering complex elements, which is detailed in the following section.

Only recover a complex element when all components can be found.

Complex header elements, such as type 7s and type 12s, specify the number of component elements that are embedded within them. Use this repair setting to correctly set the number-of-components or to delete the complex element.

When recovering elements from a partially corrupted sub-file (stream) the whole complex element may not be recoverable. This repair setting gives you the option of either recovering part of the complex header or getting rid of it.

The error number reported is 322.

Internal Data Damage

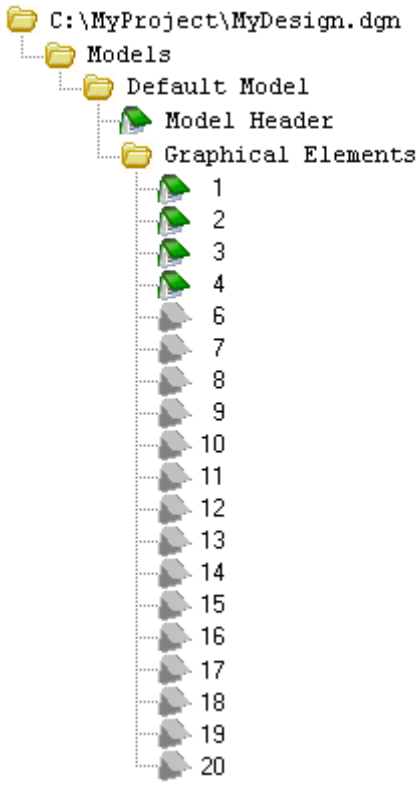
The section of the “Structured Storage” repair settings gives you a number of options for recovering elements past missing or damaged data areas. It also gives you an option for rebuilding missing or damaged model header data. The following sections describe these options in detail.

Recover elements beyond missing or damaged graphical data.

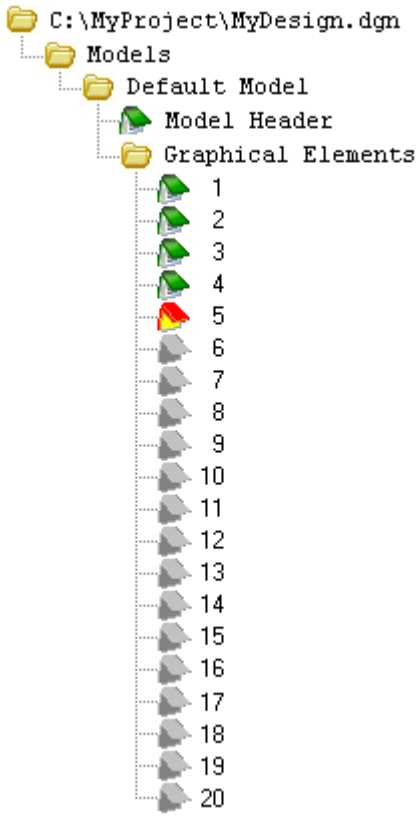
Graphical data includes the visible geometry in your design.

The graphical data area is made up of one or more sub-files (streams). MicroStation cannot read sub-files after a missing or damaged sub-file. What *FileFixer* does to fix these problems is to put an empty sub-file in place of a missing or damaged sub-file. MicroStation can then find and display the graphical elements in sub-files after a missing or damaged sub-file.

In the following illustration, sub-file 5 is missing. Elements stored in later sub-files are not displayed.



In the following illustration, sub-file 5 is damaged (unreadable). Elements stored in later sub-files are not displayed.



The error number for a damaged sub-file is 802. The search setting that detects this error is *Settings* | *Change Settings* | *Search...* | *Structured Storage* | *Damaged graphical data*.

The error number for a missing sub-file is 801. The search setting that detects this error is *Settings* | *Change Settings* | *Search...* | *Structured Storage* | *Missing graphical data*.

Recover elements beyond missing or damaged non-graphical data.

Non-graphical data includes reference file attachment elements and type 66 elements.

The non-graphical data area is made up of one or more sub-files (streams). MicroStation cannot read sub-files after a missing or damaged sub-file. What *FileFixer* does to fix these problems is to put an empty sub-file in place of a missing or damaged sub-file.

The error number for a damaged sub-file is 802. The search setting that detects this error is *Settings* | *Change Settings* | *Search...* | *Structured Storage* | *Damaged non-graphical data*.

The error number for a missing sub-file is 801. The search setting that detects this error is *Settings* | *Change Settings* | *Search...* | *Structured Storage* | *Missing non-graphical data*.

Recover elements beyond missing or damaged non-model data

Non-model data includes the design file header, view information, level tables and shared cell definitions in your design.

The non-model data area is made up of one or more sub-files (streams). MicroStation cannot read sub-files after a missing or damaged sub-file. What *FileFixer* does to fix these problems is to put an empty sub-file in place of a missing or damaged sub-file.

The error number for a damaged sub-file is 802. The search setting that detects this error is *Settings* | *Change Settings* | *Search...* | *Structured Storage* | *Damaged non-model data*.

The error number for a missing sub-file is 801. The search setting that detects this error is *Settings* | *Change Settings* | *Search...* | *Structured Storage* | *Missing non-model data*.

Rebuild missing or damaged model data.

Each V8 model (or cell) consists of

- A model *header* (stores the model's name, description, type and other model attributes).
- Non-graphical elements (such as reference file attachments and type 66 elements).
- Graphical elements (lines, circles, cells, etc.)

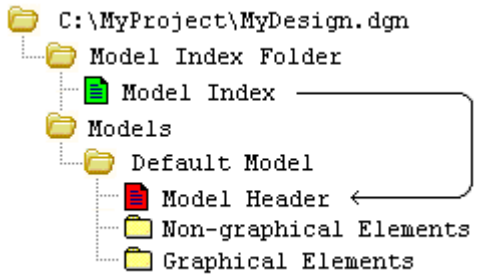
The model information (model's name, description, type, etc.) for all models is also stored in the model index.

A missing or damaged model header makes that model and often the entire file unusable.

A damaged model index makes all models and often the entire file unusable.

If Design History is present in the design file, *FileFixer* restores model header data from Design History.

FileFixer can reconstruct a missing or damaged model header using information from the model *index* (which is stored elsewhere in a V8 file).



Model Attribute	Handling
Name	Restored by *FileFixer*.
Description	Restored by *FileFixer*.
Dimension Flag (2D/3D)	Restored by *FileFixer*.
Type (Design, sheet, etc.)	Restored by *FileFixer*.
“Can be placed as a cell” flag	Restored by *FileFixer*.
Cell type (graphic or point)	Restored by *FileFixer* only if “Can be placed as a cell” setting was originally ON. Otherwise, defaults to “Graphic” (re-set to “Point” manually).
Working Units	Restored by *FileFixer* if Design History is available. Otherwise, working units must be re-set manually — *FileFixer’s* report reminds you to do this.

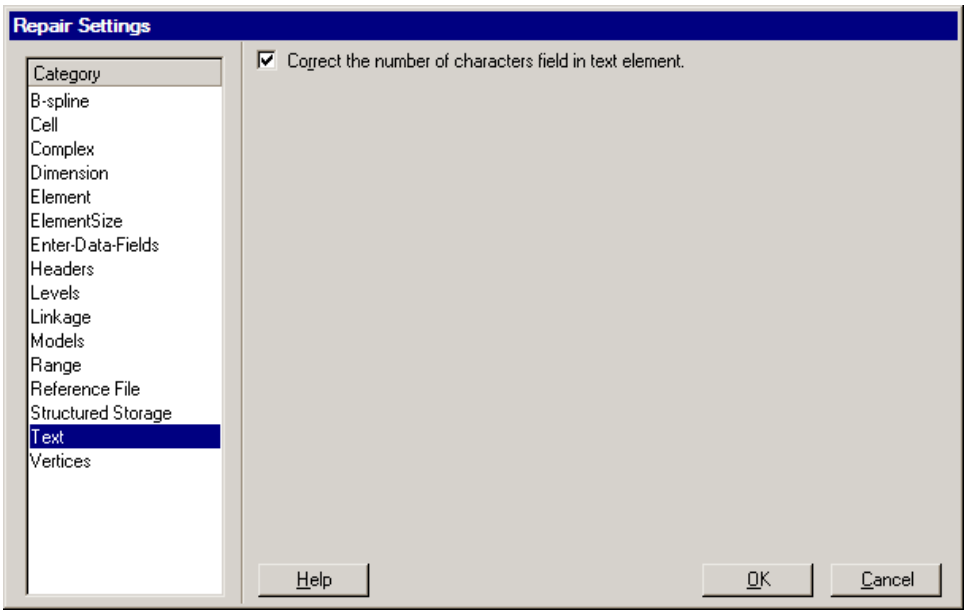
Due to the severity of this error, *FileFixer* terminates Search processing after the first instances of a damaged or missing model header is encountered.

FileFixer Repair processes *all* model headers.

A model index stream is rebuilt by using the information from all the model headers.

The error numbers reported are 801 (missing) or 802 (damaged).

Text



“Repair Settings”, “Text” category.

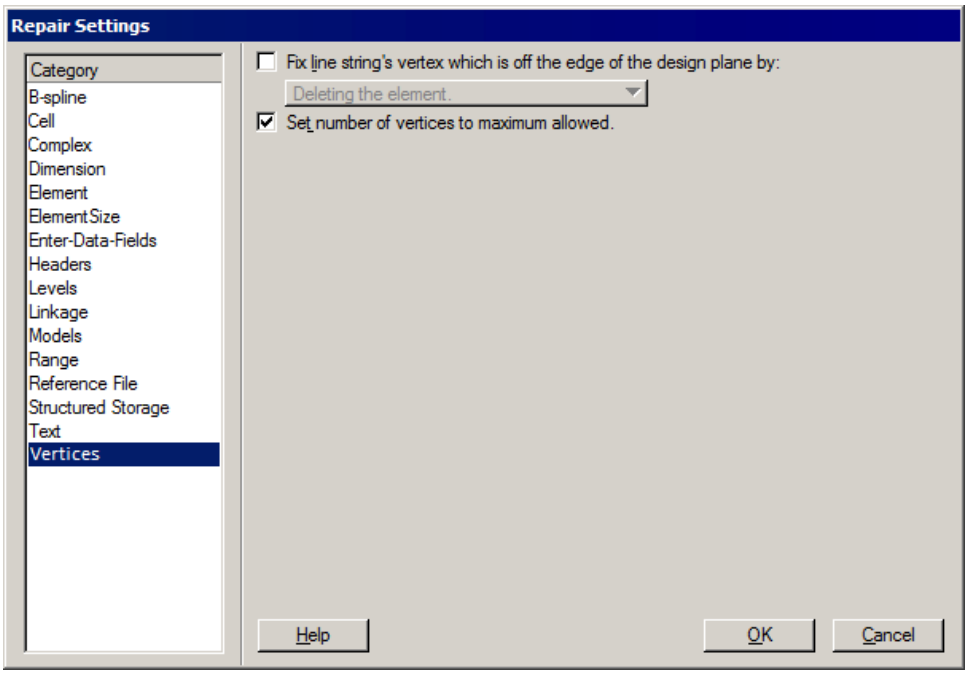
Correct the number of characters field in text element

Set the number of characters to the number of characters that actually exist.

The search setting that detects this error is *Settings* | *Change Settings* | *Search...* | *Text* | *Text string overflows the element’s length*.

The error number is 560.

Vertices



Repair Settings | *Vertices* dialog box

Fix line string's vertex which is off the edge of the design plane by

One of the string's vertices is off the edge of the design plane.

The vertex can either be adjusted or the element can be deleted. To adjust the corrupted vertex the vertices on both sides of it must be on the design plane (otherwise, the vertex cannot be adjusted).

The search setting that detects this problem is *Settings* | *Change Settings* | *Search...* | *Vertices* | *Line string's vertex is off the design plane*.

The error number reported is 373.

Set number of vertices to maximum allowed

The number of vertices property of an element is greater than the maximum allowed of 5000. This option adjusts the value to the maximum allowed.The search setting that detects this problem is *Settings* | *Change Settings* | *Search...* | *Vertices* | *Too many vertices on line string, shape, curve, etc*.

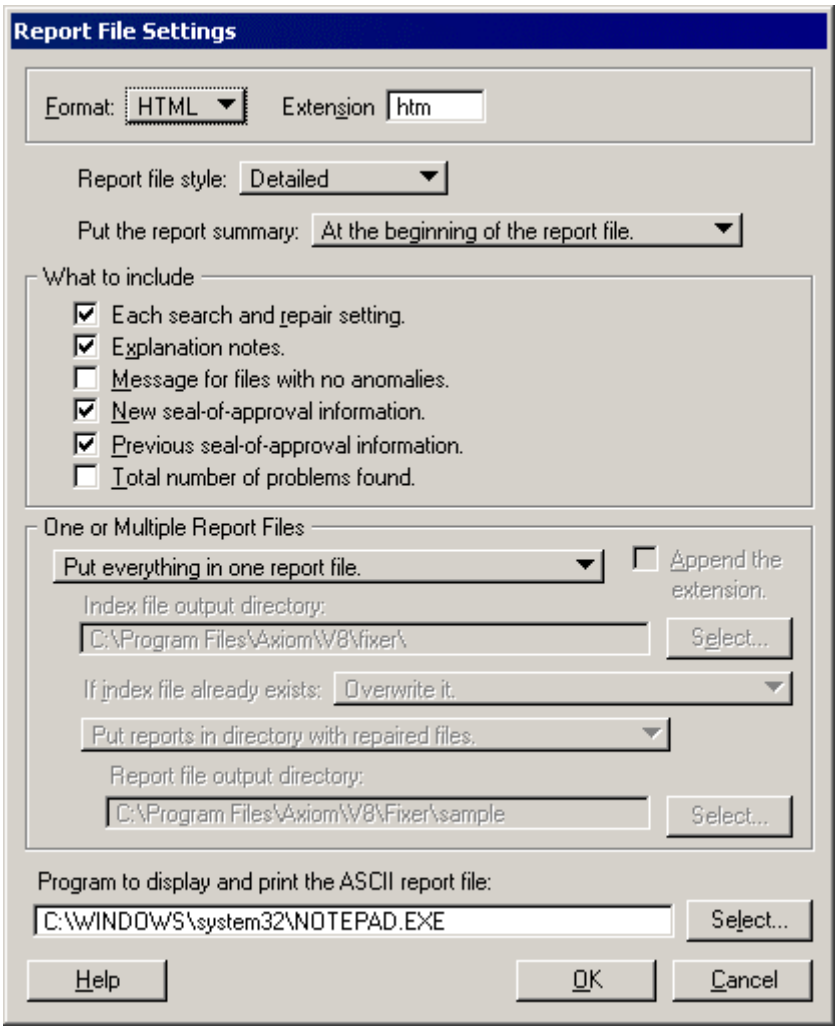
The error number reported is 787.

Chapter 10 — Report File Settings

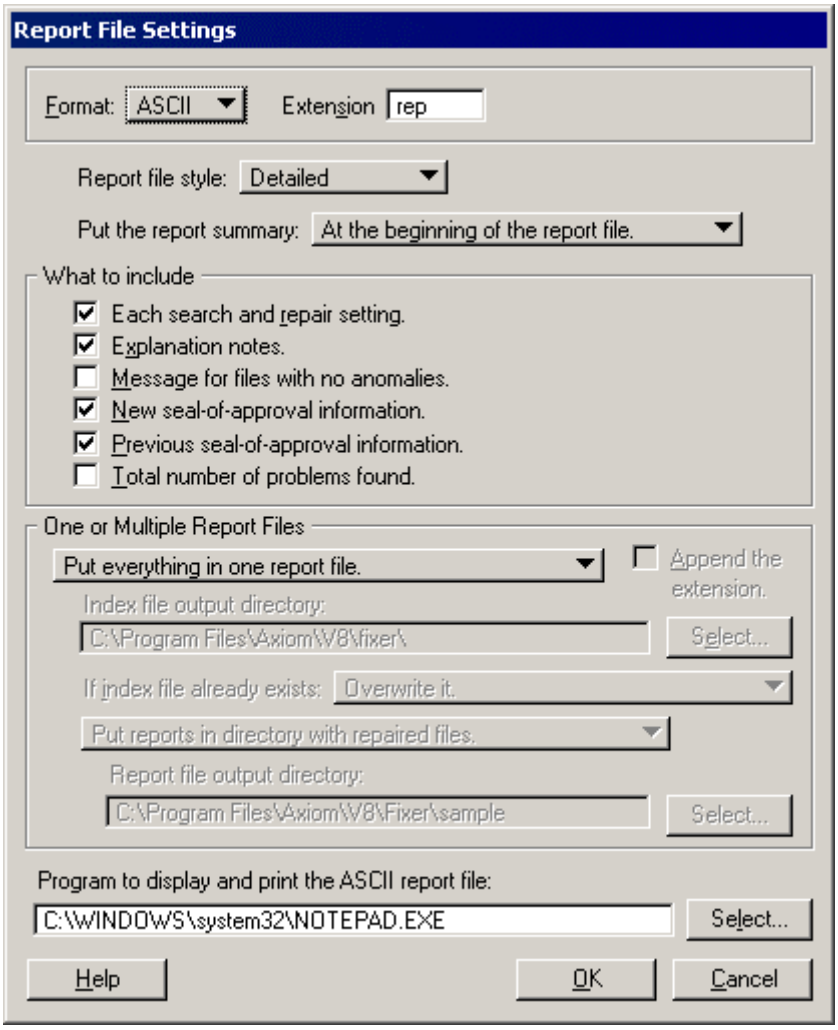
Factory defaults

Unless indicated otherwise, the settings illustrated in the following screen images reflect *FileFixer's* factory default settings.

Report File Settings dialog box



"Report File Settings" dialog box for an HTML report file



"Report File Settings" dialog box for an ASCII report file.

Report file settings, factory defaults

FileFixer generates one or more reports each time you run it. The dialog box above shows the factory default report settings. When you use these defaults you get a single report in HTML file format from each *FileFixer* run.

The default report file name and path are as follows:

When processing a single file:

The report will be created in the directory with the design file and have the same name as the design file. The default extension is "`.htm`". When you select ASCII as your output mode your extension is "`.rep`".

When processing multiple files:

When you process multiple files in an "Edited File List" or by using a wildcard, *FileFixer* creates a single report file with information about all files processed. The report will be created in the directory of the active file and be named "`fixer.htm`".

Other report default values

By default, *FileFixer* creates a summary of errors found at the top of the report.

The information from any previously existing seal-of-approval displays on the report.

If you are running in "Search for Problems" mode, a new seal-of-approval is created and the information written to it is also displayed.

FileFixer generates a list of all selected settings and some report explanation notes at the bottom of the report. This can be helpful in determining what you were running when looking at an old report.

Note: You can override automatic report file name generation by creating and setting the MicroStation configuration variable FIXER_DISABLE_AUTO_RPTFIL_NAME = "Yes". When this configuration variable is set to "Yes" *FileFixer* will use the report file name you enter (won't autogenerate report file names).

Format:

You can select HTML or ASCII format.

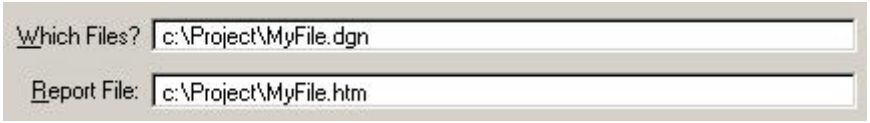
When you select "ASCII" and use .rep extension, reports will display in your computer's default text editor.

Extension

When you select ASCII format the "Report file extension:" field defaults to "rep". When you select "HTML" the extension defaults to "htm". However, you can enter any extension you want.

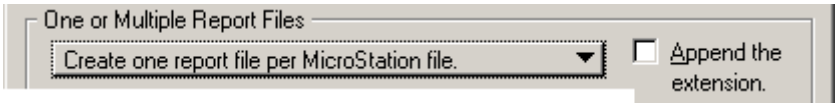
Append the extension

By default in *FileFixer's* main dialog box, the report file has the same base name as the active design file and the ".dgn" extension is replaced with ".htm".

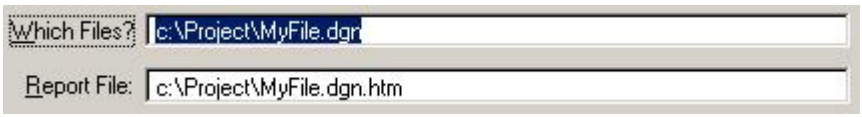


You have the option of appending report file extension to the base name plus extension of the active file. However, the option "Append the extension" is grayed out by default and is *not* available when using the "Put everything in one report file" setting (shown below):

The “Append the extension” check box becomes available when you change the “One or Multiple Report Files” setting “Create one report file per MicroStation file” (shown below):

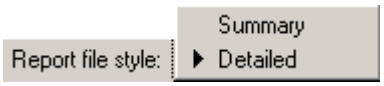


When “Append the extension” is selected, *FileFixe*r adds the extension to the active files base name plus extension (in *FileFixer’s* main dialog box):



As a pleasant reminder, the “Append the extension” option is *only* available when using the “Create one report file per MicroStation file” setting.

Report file style



By default, *FileFixer’s* “Report file style” is “detailed”. This means *FileFixer* will report each error per element. See the example below:

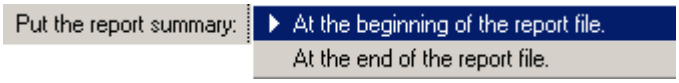
```
290, ID=3567 -- Cell "Chair".  Shared cell definition has no shared cell
instances.  [303].

2000009, ID=229 -- Reference file <Floor.dgn> is missing.  [343].

4000012, ID=1219 -- Current font library or resource has no font 10.  [144].

4000767, ID=1720 -- Text string doesn't contain any visible characters.
[146].

4002775, ID=13803 -- Phantom element. Complex bit set for non-component.
[702].
```

If you the selected “Summary” as the “Report file style” then *FileFixer* only reports the number of occurrences of each error per file, instead of listing each error per element. See the example below:

```
File: V:\v8test\Fixer\Problem V8 format files\b01.DGN


  Model: "Default" (2D)

  Description: Created from V7 conversion


  2 -- Element range doesn't match the min-max of its vertices.  [705]

  1 -- Complex header range is badly off.  [720]

  1 -- Element has too few vertices, points, boundaries, etc.  [789]


Non-model section had 129 elements. FileFixer found 0 errors in it.

Model "Default" had 228 elements. FileFixer found 4 errors in it.
```

```
1 of the 1 files analyzed are in danger.
```

Put the report summary

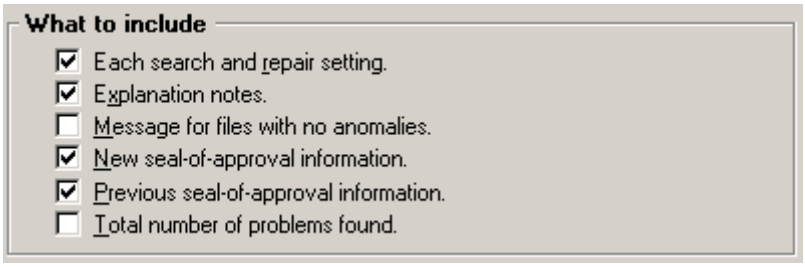


Put the following report summary information at the beginning of the report file or after all the errors for the last design file or cell library:

```
Report Summary
  3 -- Text element doesn't contain any visible characters.  [146].
  1 -- Shared cell definition has no shared cell instances.  [303].
  2 -- Complex header range is incorrect.  [320].
  1 -- Reference file is missing.  [343]
```

By default, *FileFixer* puts the report summary at the beginning of the report file.

What to include



Each search and repair setting.

When selected, the state (on or off) of each search and repair setting is included in the report file.

Explanation notes.

When this setting is checked, explanations about *FileFixer*'s error numbers, error severities and definitions for *FileFixer*-specific terminology are added to the end of the report file.

Message for files with no anomalies.

If a design file or cell library has no errors in it then the name of the design file or cell library is not printed in the report file. The report file will normally contain the names of the design files and cell libraries that have some kind of corruption in them. This setting is provided so you can see the names of all the design files and cell libraries that were searched by *FileFixer* even if they had no errors in them.

New seal-of-approval information.

The seal-of-approval contains the following information: error numbers found, date and time when *FileFixer* was run on the file, and the level of error reporting that was done (e.g. “Fatal error only”, “Errors and fatal errors”, etc.).

Previous seal-of-approval information.

If the file being processed contains a *FileFixer* seal-of-approval from a previous “Search for Problems” run, you can use this setting to display a summary of information found in that run..

Total number of problems found.

FileFixer will add the following messages to the end of the report files:

```
999 problems found
999 elements modified
```

The number of problems found and the number of elements modified will only be the count for those design files and cell libraries specified in the report file. If you are using the setting to have one report file per design file then these counts will only be for the one file, not for all the files you are processing (if more than one).

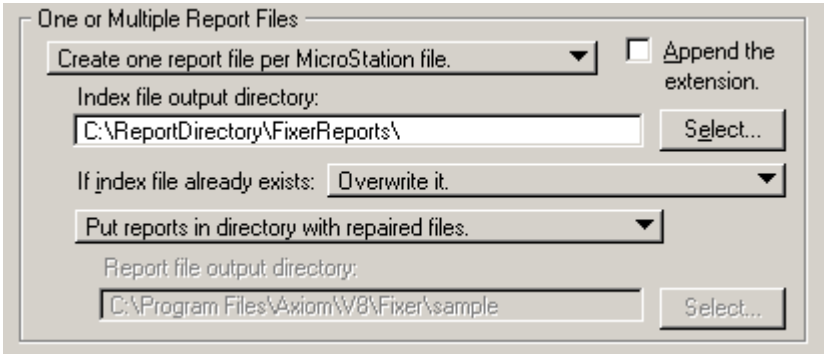
One or Multiple Report Files

When you run *FileFixer* on a large number of files, the default format of “Put everything in one report file” may be inconvenient. *FileFixer* gives you the ability to create one report file for each file processed with an index containing a summary of all errors found and links to the individual reports.

Put everything in one report file

When you select this setting, *FileFixer* builds a single report file for all design files and cell libraries processed. This is the default. The report file will be in the directory of the active file and be named “`fixer.rep`” or “`fixer.htm`”.

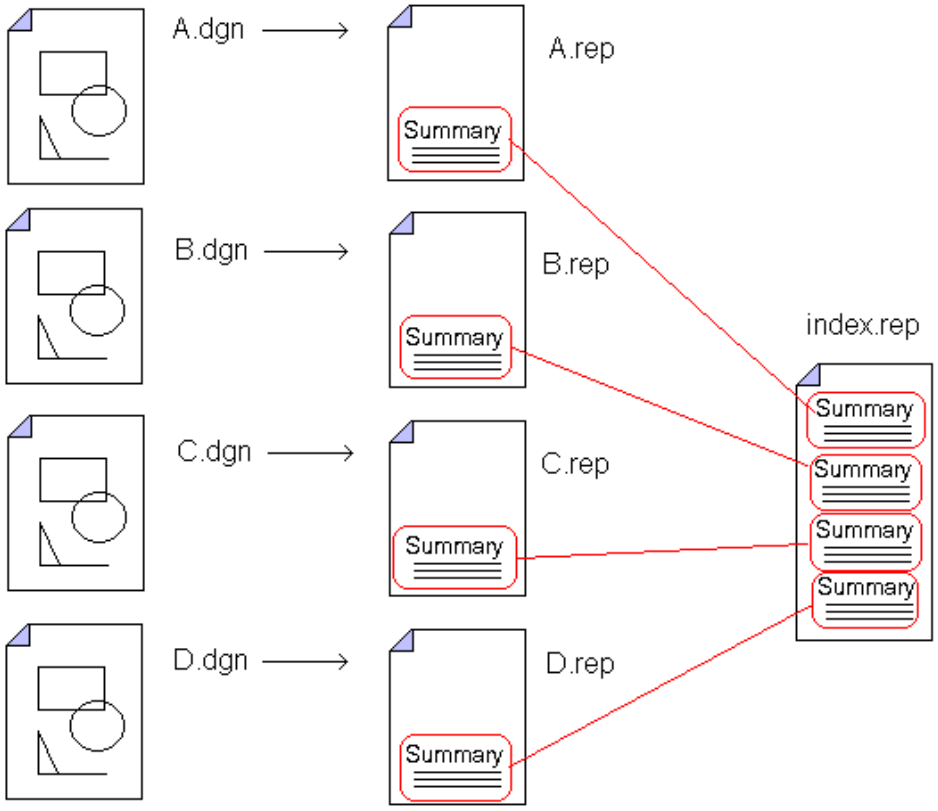
Create one report file per MicroStation file



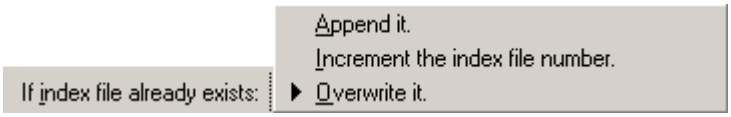
When you select “Create one report file per MicroStation file.”, *FileFixer* builds one report file for each design file and cell library processed. It also creates an “index” file containing a summary for each file processed.

Index file output directory

When you build one report file per MicroStation file, *FileFixer* creates an index file containing a copy of the report summary for each design file processed. (See image below.) When you use HTML format, the index file will also include a hyperlink to each individual, detailed report file.



If index file already exists



If the index file already exists then you have three options, (1) append the new report summaries at the end of the index file, (2) increment the index file number, or (3) overwrite the old index file with the new one.

The format for the index file name is “index9999.rep” or “index9999.htm” when you are incrementing the index file number, otherwise, it is just “index.rep” or “index.htm”. *FileFixer* allows up to 9,999 index files when you are incrementing the index file number.

Directory for individual reports

When you select “Create one report file per MicroStation file”, the report files can be placed in one of the following locations:

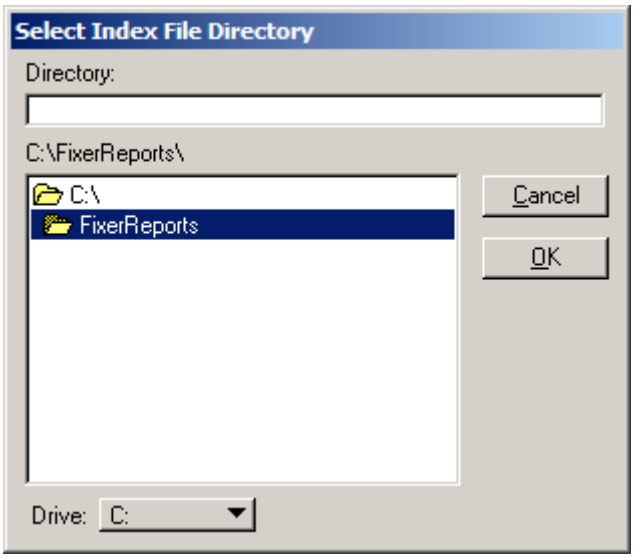
Put reports in the specified directory.
Put reports in directory with repaired files.
Put reports in directory with backup files.
Put reports in directory with original files.

Your selections in the “When repairing a design file or cell library” section of the *Settings* | *Change Settings* | *General...* dialog box determine when the above settings are enabled or disabled.

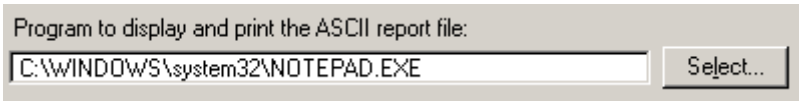
If one of the above settings is specified, then the “Report file:” <Select...> button on the main *FileFixer* dialog box will be disabled. For example:

Put reports in the specified directory.

If you select “Put reports in the specified directory” then you need to also specify the directory path for the report files. You can do this by entering the directory in the “Report file output directory:” field or by pressing <Select>. After you press <Select> the following dialog box opens and allows you to specify a directory. The main *FileFixer* dialog box “Report File:” field shows the selected directory followed by a *.rep or *.htm.



Program to display and print the ASCII report file:

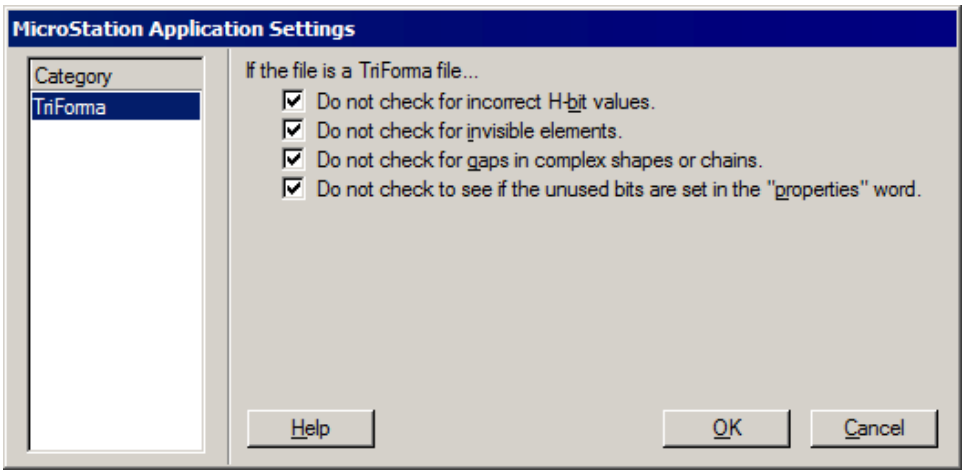


This field defaults to notepad.exe, but you can use the <Select...> button to choose a different text editor.

Chapter 11 — MicroStation Application Settings

MicroStation Applications Settings dialog box...

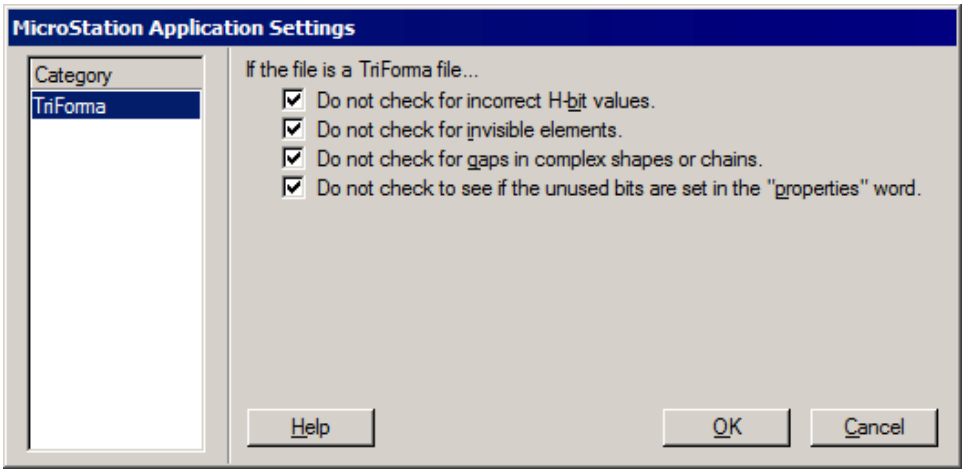
MicroStation Application Settings pertain to *specific* engineering applications — just TriForma for now.


MicroStation Application Settings
Category
TriForma
If the file is a TriForma file...
Do not check for incorrect H-bit values.
Do not check for invisible elements.
Do not check for gaps in complex shapes or chains.
Do not check to see if the unused bits are set in the "properties" word.
Help
OK
Cancel

TriForma settings

Sometimes TriForma creates elements with characteristics that are not normally found in non-TriForma files.

The settings in this dialog optimize the information *FileFixer* reports about TriForma files.


MicroStation Application Settings
Category
TriForma
If the file is a TriForma file...
Do not check for incorrect H-bit values.
Do not check for invisible elements.
Do not check for gaps in complex shapes or chains.
Do not check to see if the unused bits are set in the "properties" word.
Help
OK
Cancel

The options in this dialog correspond with *FileFixer* error numbers 383, 304, 151 and 132 respectively — each of these errors is described fully in the Search Settings and Repair Settings sections of this guide.

These settings are automatically activated when *FileFixer* detects TriForma elements in a design file.

These settings are ignored when the file being processed is not a TriForma file.

Chapter 12 — Reports

Report Summary

FileFixer always generates a report that summarizes the actions taken during each processing run.

The first “landmark” to check in a report file is the Report Summary. By default, this summary is at the top of the report.

Your report Summary might look like this:

```
Report Summary
  No errors were found.
```

If so, you’re done!

Otherwise, your Report Summary indicates what is reported in the remainder of the report file. The number on the left in the “Report Summary” section is the number of times that an element with this type of error was encountered and is the total for all files that were processed in that run:

```
Report Summary
    14 -- Text uses font that is not in current font library.  [144].
     3 -- Text element doesn't contain any visible characters.  [146].
    12 -- Shared cell definition has no shared cell instances.  [303].
     2 -- Reference file is missing.  [343].
    10 -- Phantom element. Complex bit is set for non-component.  [702].
```

Usually the best strategy is to find the highest (most severe) error number and use the *Edit* | *Find* feature of Notepad or your Browser to locate instances of the most severe error in your report. Such a search will usually jump to the detail portion of the report file. Here’s an excerpt from an actual report:

```
290, ID=3567 -- Cell "Chair".  Shared cell definition has no shared cell
instances.  [303].
2000009, ID=229 -- Reference file <Floor.dgn> is missing.  [343].
4000012, ID=1219 -- Current font library or resource has no font 10.  [144].
4000767, ID=1720 -- Text string doesn't contain any visible characters.
[146].
4002775, ID=13803 -- Phantom element. Complex bit set for non-component.
[702].
```

As in the above excerpt, each detail line in your reports will contain the following:

File position number, ID=element ID -- Error message. [Error number].

Each of these four parts is described below.

The File Position Number

The left-most number is the *file position* of the element. Here's how to interpret file position numbers for an element:

File position	Meaning
0–1,999,999	The element resides in the *non-model* section of a V8 file. The non-model section contains information which is shared by all models in the file. There is only **one** non-model section in each V8 design file.
2,000,000–3,999,999	The element resides in the *control* section of a model. Each control section contains non-graphical information for a model, such as settings and reference file attachments. There can be **many** control sections in a V8 design file, one or more for each model.
4,000,000 and above	The element resides in the graphical section of a model. The graphical section contains the *visible* elements in a model. There can be **many** graphical sections in a V8 design file, one or more for each model.

File position numbers are *not* unique within a V8 file. For example, the first graphical element in each model has file position 4,000,000.

Further, an element's file position can change when a V8 file is saved. File position numbers are not permanent.

File position is included in *FileFixer* reports because they are useful for identifying the element's role in a V8 design file (as in the above table).

Note: The numerical ranges in the table above are arbitrary and subject to change by Bentley in the future.

The Unique Element ID

The element ID follows the file position number in each detail line of the above example.

The element's ID is unique within a file.

Let's examine the above report detail lines with a microscope:

File Position	Element ID	Comments
290	ID=3567	This element's file position is less than 2,000,000, which means it resides in the non-model section. We can tell from *FileFixer*'s error message (see above) that this element is a

		shared cell definition — shared cell definitions *should* be shared amongst models, so it makes sense that this element resides in the non-model section of the file. This particular shared cell definition has element ID = 3567.
2000009	ID=229	This element's file position indicates it is in the control section of a model. We can tell from *FileFixer*'s error message (see above) that this element is a reference file attachment. A reference file attachment element is not itself visible, so it makes sense that this element resides in the *control* section of a model. The reference file attachment element's ID = 229.
4000012	ID=1219	This element's file position indicates it is in the graphical section of a model. We can tell from *FileFixer's* error message (see above) that this is a text element. A text element is visible, so it makes sense that this element resides in the graphical section of a model. The text element's ID = 1219.

Element IDs do not necessarily correspond to the order of elements in a file.

Note: You can learn a lot more about the elements in a V8 file using Axiom's *Problem Element Viewer*.

The *FileFixer* Error Message and Error Number

In each detail line of a *FileFixer* report (as in the above example), the file position and element ID are followed by a *FileFixer* error message and an error number in square brackets.

The error numbers are described in the "Error Messages" section of this guide.

Chapter 13 — Error Messages

Operational errors and messages

Operational errors are reported with error numbers in the range 000–099. These messages concern something that should be brought to your attention.

003 — Could not open <_____>.

Cause — *FileFixer's* structured storage search settings are disabled, but this file has structured storage errors, such as damaged sub-folders or sub-files.

Cause — You don't have the privilege to create files in the directory you specified.

Solution — Have your system administrator find out why the file couldn't be created and handle.

Solution — Contact Axiom to find out what is wrong with the structured storage of this file.

004 — Process aborted by user.

Cause — User pressed <Esc> during *FileFixer* processing.

005 — Too many errors have already been reported for this file.

Cause — The "Search for Problems" command lets you specify the maximum number of errors to report for a file. *FileFixer* identified more than the maximum number of errors in this file.

Solution — Repair the file or increase the maximum number of errors allowed.

016 — Specified file does not exist.

Cause — User typed in a directory path and/or file name that does not exist.

Cause — Non-existent file name follows /L option (in "Which Files?" field).

Effect — *FileFixer* cannot process the specified file.

Solution — Check *FileFixer*'s report for the file name which could not be found.

017 — Input file <_____> cannot be opened.

Cause — You do not have the necessary privilege to open this file.

Cause — Someone else or another process has the file open at the moment.

Cause — There is something wrong with the computer or file system.

Effect — This file cannot be processed by *FileFixer* until the problem is corrected.

Solution — Have your system administrator determine why the file cannot be opened.

018 — Model information could not be read so this design file cannot be processed.

Cause — The model index is missing or damaged.

Effect — This file cannot be processed by *FileFixer* until the problem is corrected.

Solution — Try opening and closing the file in MicroStation V8, then re-run *FileFixer*.

019 — This design file has digital protection and will be skipped.

Cause — This V8 file is digitally protected.

Effect — *FileFixer* does not process protected files.

```
File: c:\MyProject\MySecureFile.dgn
  *** -- This design file has digital protection and will be skipped.  [019].
```

Solution — Unprotect your file, then re-run *FileFixer*.

021 — Type 9 (design file header) has been replaced.

Cause — You directed *FileFixer* to replace the type 9 design file header element (the first element in the file) with a type 9 from another design file. This message is reported for the replaced element.

Effect — The type 9 from the donor file you specified using the *Repair...* | *Header* | *Replace the design file header element* will replace this element.

022 — Non-model table has been replaced.

Cause — You directed *FileFixer* to replace a non-model table with a non-model table from another design file. This message is reported for the replaced element.

Effect — The non-model table from the donor file you specified using the *Repair...* | *Header* | *Replace Non-Model Header or Table* will replace this table.

029 — Backup file already exists.

Cause — You are using “Work on Original” mode and a .bkp version of this file already exists. *FileFixer* will not overwrite a .bkp file.

Effect — *FileFixer* will continue to protect your files (won’t overwrite) until this is remedied.

Solution — Rename the .bkp file to something else.

Solution — Use wildcards in the backup file extension. For example, “.bk?”.

Solution — Turn “Work on Original” mode OFF. That way *FileFixer* will create a *copy* of the original file (with a .fix extension) and process the copy, retaining the original untouched.

Solution — If you are absolutely positive that this backup file will never be needed, you can delete it.

050 — Error writing element to output (.fix) design file.

Cause — In most cases this error is caused by insufficient disk space on the disk to which *FileFixer* is writing the output design file.

Effect — The output design file will not be complete.

Solution — Copy the design file you want to repair to a disk with more disk space.

Solution — Free up some room on the disk in question by deleting unnecessary files.

Solution — If the above doesn’t solve this, contact Axiom Support for help at support@AxiomInt.com.

060 — This isn't a V8 design file.

Cause — The selected file is not a V8 design file.

Effect — *FileFixer for V8* is designed to process V8 files only.

Solution — Use the V7 version of *FileFixer* to process V7 files.

062 — This is the first element indicating that this file contains _____ attribute linkage.

Cause — This design file contains at least one element with application-specific attribute linkage. This is not a corruption. As *FileFixer for V8* evolves, various application-specific settings may be added. As of this writing, TriForma-specific settings are supported. This message indicates that at least one TriForma element has been detected in the file — so *FileFixer* automatically activated TriForma-specific settings when processing the file.

Example:

```
4000000, ID=43676 -- This is the first element indicating this file contains
TriForma attribute linkage.  [62].
```

Application-specific settings are displayed on the *Settings* | *Change Settings* | *MicroStation Applications dialog*.

Suspicious Conditions

Suspicious conditions are reported with error numbers in the range 100–199. There is something not quite right about the reported elements.

103 — Reserved bits are set in the element header.

Cause — Bits 10:15 are unused in the second word of every MicroStation V8 element. This allows for future enhancements to the element format. These bits should be clear (set to 0). If they are not, this message is generated.

Technically this is not an error because there are no known programs that rely on these bits at this time. However, when trying to pick up the trail of a valid element chain, the sophisticated algorithm *FileFixer* uses to automatically repair corrupted design files will make note of the fact that a potentially valid element has set reserved bits and severely downgrade the likelihood that this element is part of a valid element chain.

Effect — None known at this time.

Solution — Use *FileFixer's* "Automatic Design File Repair" command with the *Settings* | *Change Settings* | *Repair...* | *Element* | *Fix reserved bits by clearing them* repair setting turned on to repair the element. This setting is OFF by default.

119 — Element is too close to design file edge.

Cause — According to its range block, this element is located dangerously close to the edge of the design plane (or design cube).

Cause — The origin of an ellipse, arc, text or text node is dangerously close to the edge of the design plane (or past the edge).

Cause — A curve's beginning and ending slope points are too close to the edge of the design plane.

Effect — Element may be unselectable, unmovable or excluded from fence operations. You may experience anomalies with actions such as plotting and Save As.

Solution — Use *Problem Element Viewer* to see what this element looks like and where it is located. If this is a stray, unwanted element and is actually causing plotting (or other) problems, then use *Problem Element Viewer* to delete it.

Note: *Problem Element Viewer* deletes elements using MicroStation's delete command. If MicroStation is not able to delete an element which manifests error 119, use the solution below.

Solution — Add "119" to the "Delete elements with the following error numbers" list under *Settings* | *Change Settings* | *Repair...* | *Element* to have *FileFixer* delete such elements in "Automatic Design File Repair" mode.

Note: It is also possible for an element to contain range values which are beyond the edge of the design plane (or design cube). This problem is new for V8 — in V7 it is not possible to *exceed* the design edge. See description of error 619 for more information.

Note: V8 element coordinates range from 4,503,599,627,370,495 (max) to -4,503,599,627,370,496 (min). [From …\Program\MicroStation\mdl\include\msdefs.h.]

132 — Reserved bits are set in the “properties” word.

Cause — The property word of every graphic element and complex header contains four bits which are not used for anything and which should always be zero. The reserved bits are bits 4:6 and 8 of the properties word.

Effect — MicroStation does not use this information yet, so this error should not cause any difficulties. However, the possibility exists that future enhancements to MicroStation could start using this information, in which case, the non-zero status of this information could cause as yet indeterminable errors.

Solution — None needed at this time.

144 — This font library does not contain font ____.

Cause — The font library MicroStation is currently using does not contain the font used by this text element. Either this element is corrupt or the font library MicroStation is using is corrupted.

Effect — MicroStation will display this element using its default font.

Solution — Find and attach the missing font library.

Solution — Change this element’s font to one that you *do* have in your font library.

146 — Text element doesn't contain any visible characters.

Cause — This element doesn’t contain any visible characters, it may contain only spaces.

Effect — This element does not violate any MicroStation design file construction rules.

Solution — Enter 146 into the string of error numbers in the text field for the *Settings | Change Settings | Repair... | Elements | Delete problem elements based on the error they generate* repair setting, and then use *FileFixer*’s “Automatic Design File Repair” command to delete elements with this error.

151 — Gap found in 2D complex shape or chain.

Cause — When creating a 2D complex shape or chain from multiple linear elements that have gaps between the endpoints, MicroStation will not actually create the connecting lines, but MicroStation will *display* a line between those endpoints.

Effect —If these 2D complex elements are later dropped, gaps will exist between the component elements.

Solution — Use *FileFixer’s* “Automatic Design File Repair” command with the *Settings | Change Settings | Repair... | Complex | Add line to bridge gap in 2D complex shape or chain.* This setting is OFF by default. *FileFixer* will only create a line to fill the gap in a 2D model.

168 — Shared cell instance dependency link is pointing to a shared cell definition with a different cell name.

Cause — A shared cell has a dependency linkage containing an element ID that points to a shared cell definition that uses a different name than the shared cell instance.

Effect — This condition is allowed in MicroStation but usually indicates a corrupt dependency linkage on the shared cell instance.

Solution — Use *FileFixer*'s "Automatic Design File Repair" with the *Settings* | *Change Settings* | *Repair...* | *Linkage* | *Change the name of the shared cell instance, with dependency linkage, to match the shared cell definition's name if they don't match* setting to repair this.

169 — Non-zero Z range found in a 2D element.

Cause — One of the Z range value is non-zero.

Effect — Systems which assume the Z range of a 2D element will be zero may behave unexpectedly when they encounter this element. For example, elements may be displayed or plotted too small or cells may be placed too small.

Solution — Use *FileFixer*'s "Automatic Design File Repair" with the *Settings* | *Change Settings* | *Repair...* | *Repair range of displayable elements* setting on to repair this element. This repair setting is ON by default.

185 — Element size, excluding lines and line strings, is zero in all dimensions.

Cause — The size of the element in all dimensions is zero.

Effect — You may not be able to see this element.

Solution — Enter 185 into the string of error numbers in the text field for the *Settings* | *Change Settings* | *Repair...*| *Elements* | *Delete problem elements based on the error they generate* repair setting, and then use *FileFixer*'s "Automatic Design File Repair" command to delete elements which generate this error.

191 — Text element has more EDFs than characters.

Cause — A text element with more EDFs (Enter-Data-Fields) than characters has been encountered.

Solution — Use *Problem Element Viewer* to find out what this element looks like. Delete it. Redraw it.

Solution — Use *FileFixer's* "Automatic Design File Repair" command with the *Settings* | *Change Settings* | *Repair...* | *Enter-Data-Fields* | *Correct EDF/text length mismatches* repair setting turned on to repair the element. This setting is ON by default.

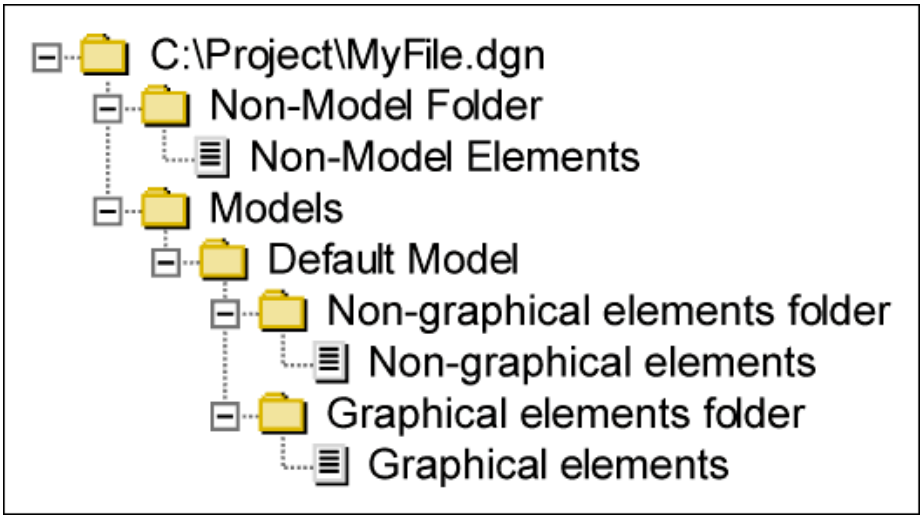
V8-specific Informational Messages (Errors 201–267)

V8-specific informational messages are reported with numbers in the range 200–299.

Structured Storage

Beginning with MicroStation V8, design files contain multiple internal folders and files, similar to Microsoft Excel or Word documents. This internal organization is called "structured storage". Documents structured this way are called "compound documents".

There are specific internal folders and files which are standard and expected in V8 design files. For example:



The errors in this section reflect missing structured storage components.

There is no search setting corresponding to these errors. *FileFixer* checks for structured storage problems automatically in every file processed because a fatal structured storage error indicates a vital portion of your file is missing or cannot be accessed — which means it is also impossible to access or analyze the elements contained within.

Note: Sub-*folders* within a compound file are called "storages". Sub-*files* within a compound file are called "streams".

201 — Missing non-graphical storage for model <_____>.

Cause — There are no control elements in this model. Control elements consist of reference file attachment elements and type 66 elements.

Effect — Suspicious for a model that contains graphical elements.

Solution — None needed at this time.

202 — Missing non-graphical data for model <_____>).

Cause — This model is missing control elements. Control elements consist of reference file attachment elements and type 66 application elements.

Effect — Suspicious for a model that contains graphical elements.

Solution — Use *FileFixer's* "Automatic Design File Repair" command with the *Settings | Change Settings | Repair... | Structured Storage | Recover elements beyond missing or damaged non-graphical data* repair setting turned on to fix. This setting is OFF by default.

203 — Missing graphical storage for model <_____>.

Cause — There are no graphical elements in this model.

Effect —Legal, but suspicious.

Solution — Delete unneeded empty models using MicroStation V8's *File | Models* dialog.

204 — Missing graphical data for model <_____>.

Cause — This model contains no graphical elements.

Effect — Legal, but suspicious.

Solution — Use *FileFixer's* "Automatic Design File Repair" command with the *Settings | Change Settings | Repair... | Structured Storage | Recover elements beyond missing or damaged graphical data* repair setting turned on to fix. This setting is OFF by default.

Note: Error 204 is suppressed for the Default model in files with a ".cel" extension because cell libraries commonly have an empty Default model.

205 — Missing non-model data.

Cause — The non-model section of the design file is missing some data.

Effect — If shared cell definitions are missing, shared cell instances will not display (or plot). If you select the *area* where the shared cell instance *should* appear and select MicroStation's Element Information command, the type 35 shared cell header might display, but its component geometry won't be listed (undefined).

Effect – If the level table is missing, levels might not appear in MicroStation's Level Manager. MicroStation's Element Information command might display a blank (empty) level name for selected elements. Level operations might unexpectedly exclude elements. For more, see description of error 415.

Effect – Other non-model data (fonts, lines styles, dimension styles, etc.) may be missing from the design.

Solution — Use *FileFixer's* "Automatic Design File Repair" command with the *Settings | Change Settings | Repair... | Structured Storage | Recover elements beyond missing or damaged non-model data* repair setting turned on to fix. This setting is OFF by default.

230 — Missing model index storage.

Cause — No model index information.

Effect — *FileFixer* cannot process this file.

Solution — Try opening and saving the file in MicroStation V8, then re-run *FileFixer*.

231 — Missing model index data.

Cause — No model index information.

Effect — *FileFixer* cannot process this file.

Solution — Try opening and saving the file in MicroStation V8, then re-run *FileFixer*.

235 — "<Element type>. TriForma is required for element modification.

Cause — TriForma has intentionally set this element so it can't be modified by MicroStation when TriForma is not loaded.

Effect —MicroStation, without TriForma loaded, will not allow these elements to be modified.

Solution — Use *FileFixer's* "Automatic Design File Repair" command with the *Settings | Change Settings | Repair...| Linkage | Clear "application required for element modification" flag in TriForma linkage* to either make these elements visible or delete them. This repair setting is OFF by default.

240 — Element type ____ is invisible.

Cause — MicroStation V8 has the ability to make elements invisible.

Effect — These invisible elements are not displayed in MicroStation's GUI and cannot be plotted. This is a new MicroStation feature and is not, strictly speaking, an element corruption but is suspicious and possibly unwanted.

Certain applications, such as Cimmetry's AutoVue, do not recognize this new element attribute and display invisible elements. This may be confusing when compared to MicroStation's display.

Solution — Use *FileFixer's* "Automatic Design File Repair" command with the *Settings | Change Settings | Repair...| Element | Fix invisible elements by:* to either make these elements visible or delete them. This setting is OFF by default.

250 — Model is locked.

Cause — MicroStation V8 has the ability to lock models.

Effect — You cannot add, modify or delete elements in a locked model. MicroStation V8.5 displays an error message "file is read-only" (which is not necessarily true). With MicroStation XM, element manipulations fail without explanation.

Solution – Use MicroStation's "model unlock" key-in to unlock the model.

Solution — Use *FileFixer's* "Automatic Design File Repair" command with the *Settings | Change Settings | Repair...| Models | Unlock models* setting enabled. This setting is OFF by default.

267 — Unused reference file level table.

Cause — A reference file level table exists, but none of the attached reference files use it.

Effect — This reference file level table is just taking up space in the design file.

Solution — Use *FileFixer's* "Automatic Design File Repair" command with the *Levels | Delete unused reference file level tables* setting enabled. This setting is OFF by default.

Minor Errors

Minor errors are reported with error numbers in the range 300–399. The reported elements may eventually cause problems.

301 — The number of words of attribute data is not a multiple of 4.

Cause — Database linkages should always be created in multiples of 4 words. The number of words in the linkage area of this element is not a multiple of 4.

Effect — May prevent database access.

Effect — May cause problems for applications that process the user linkage.

Solution — Use *Problem Element Viewer* to locate this element. Then redraw or delete the element.

302 — Type 1 (cell library header) elements are not allowed in design files.

Cause — Type 1 cell library headers (V7 only) should not be found in a V8 design file.

Effect — None known.

Solution — Use *FileFixer's* "Automatic Design File Repair" command with the *Settings | Change Settings | Repair...| Element | Remove illegal element types* setting turned on. This setting is ON by default.

303 — Shared cell definition with no shared cell instance.

Cause — A shared cell definition has been found, but there is no shared cell instance that uses that definition. This happens quite often when importing DWG files into MicroStation.

Effect —MicroStation will ignore this shared cell definition.

Solution — Use *FileFixer's* "Automatic Design File Repair" command with the *Settings | Change Settings | Repair... | Cell | Delete shared cell definition with no shared cell instances* setting turned on. This setting is OFF by default.

304 — H-Bit should not be set for this element.

Cause — According to the strictest rules of element construction, the only elements whose H-Bit should ever be set are cells (type 2), lines (type 3), shapes (type 6), complex shapes (type 14), ellipses (type 15), and element types 21 through 28. The reported element was not one of these types, yet its H-Bit was set.

Effect — May cause display or plotting problems.

Solution — Use *FileFixer's* "Automatic Design File Repair" command with the *Settings* | *Change Settings* | *Repair...* | *Element* | *Turn off illegal H-bits* setting turned on. This setting is ON by default.

Note: The H-Bit is called the H-Bit because for many element types it indicates the element is a *hole*, not solid. The H-bit has various other meanings for element types which are not in the above list.

305 — Element range doesn't match the min-max of its vertices.

Cause — This error is essentially similar to error 705, but is less severe in nature. This error is generated when the range is off by more than the user-specified tolerance, but within the tolerance -70% and 900%. See error number 705 below for more information.

Solution — Use *FileFixer*'s "Automatic Design File Repair" with the *Settings* | *Change Settings* | *Repair...* | *Repair range of displayable elements* setting on to repair this element. This repair setting is ON by default.

307 — B-spline pole element doesn't match the min-max of its poles nor its b-spline surface header.

Cause — This error is essentially similar to error **707**, but is less severe in nature. This error is generated when the range is off by more than the user-specified tolerance, but within the tolerance -70% and 900%. See error number **707** below for more information.

Solution — Use *FileFixer*'s "Automatic Design File Repair" with the *Settings* | *Change Settings* | *Repair...* | *Repair range of displayable elements* setting on to repair this element. This setting is ON by default.

309 — Type __ header misses type __ component by __ %.

Cause — This error is essentially similar to error **709**, but is less severe in nature. This error is generated when the range is off by more than the user-specified tolerance, but within the tolerance -70% and 900%. See error number **709** below for more information.

Solution — Use *FileFixer*'s "Automatic Design File Repair" with the *Settings* | *Change Settings* | *Repair...* | *Repair range of complex header elements* setting on to repair this element. This setting is ON by default.

312 — Range of right circular truncated cone is incorrect.

Cause — This error is essentially similar to error **712**, but is less severe in nature. This error is generated when the range is off by more than the user-specified tolerance, but within the tolerance -70% and 900%. See error number **712** below for more information.

Solution — Use *FileFixer*'s "Automatic Design File Repair" with the *Settings* | *Change Settings* | *Repair...* | *Repair range of displayable elements* setting on to repair this element. This setting is ON by default.

314 – Duplicate reference file slot number found.

Cause — This element is a reference file attachment whose slot number matches a previous reference file attachment in this model.

Effect — Only the first reference file attachment with this slot number will display in MicroStation's References dialog. Other attachments using the same slot number won't display and can't be deleted from MicroStation's References dialog.

Effect — We added this new search option because customers reported duplicate slot numbers prevented successful processing of attachments by third party applications.

Solution — Run *FileFixer's* "Automatic Design File Repair" with the *Settings* | *Change Settings* | *Repair…* | *Reference File* | *Set slot number to an unused slot number*. This setting is ON by default.

320 — Complex header range is off by ___%.

Cause — This error is essentially similar to error 720, but is less severe in nature. This error is generated when the range is off by more than the user-specified tolerance, but within the tolerance -70% and 900%. See error number 720 below for more information.

Solution — Use *FileFixer*'s "Automatic Design File Repair" with the *Settings* | *Change Settings* | *Repair...* | *Repair range of complex header elements* setting on to repair this element. This setting is on by default.

322 — Component count in complex header is incorrect.

Cause — Complex header elements, such as text nodes (type 7). cells (type 2 and 34) and strings (type 12), specify the number of component elements that are embedded within them. *FileFixer for V8* detected that the *actual* number of components does not match the count stored in the complex header.

Effect — MicroStation V8 may not be able to load the design file.

Solution — Use *FileFixer's* "Automatic Design File Repair" command with the *Settings* | *Change Settings* | *Repair...* | *Structured Storage* | *Only recover a complex element when all components can be found* turned on (ON by default) to correct the number-of-components.

323 — A deleted element was found within a complex element. .

Cause — Complex elements should not contain deleted elements.

Effect — Unknown.

Solution — Run *FileFixer's* "Automatic Design File Repair" command with the *Settings | Change Settings | Repair… | Complex | Handle illegal components in complex element.* setting turned on. With this setting turned on you have the choice of dropping status on the complex header or squeezing out the deleted elements. This setting is on by default and its default setting is to squeeze out deleted elements.

324 — Range of arc or ellipse is incorrect.

Cause — This error is essentially similar to error **724**, but is less severe in nature. This error is generated when the range is off by more than the user-specified tolerance, but within the tolerance -70% and 900%. See error number **724** below for more information.

Solution — Use *FileFixer*'s "Automatic Design File Repair" with the *Settings | Change Settings | Repair... | Repair range of displayable elements* setting on to repair this element. This setting is on by default.

325 — Range of text element is incorrect.

Cause — This error is essentially similar to error **725**, but is less severe in nature. This error is generated when the range is off by more than the user-specified tolerance, but within the tolerance -70% and 900%. See error number **725** for important information.

Solution — Use *FileFixer*'s "Automatic Design File Repair" with the *Settings | Change Settings | Repair... | Repair range of displayable elements* setting on to repair this element. This setting is on by default.

Be sure you have the correct font library specified in MicroStation.

327 — Range of text node element is incorrect.

Cause — This error is essentially similar to error 727, but it is less severe in nature. This error is generated when the range is off by more than the user-specified tolerance, but within the tolerance –70% and 900%.

Effect — Possible element snapping, selection, and fence manipulation problems.

Solution — Use *FileFixer*'s "Automatic Design File Repair" with the *Settings | Change Settings | Repair... | Repair range of displayable elements* setting on to repair this element. This setting is on by default.

333 — Class of element is illegal.

Cause — The class of this element is undefined (greater than 6).

Effect — This element may fail to display, may fail to plot or may be left out of some fence operations.

Solution — Use *FileFixer's* "Automatic Design File Repair" command to set the class to a legal value. Remember to turn on the *Settings* | *Change Settings* | *Repair...* | *Element* | *Set illegal class to* ___ repair setting as it is OFF by default.

336 — A component element is on a different level than its complex header.

Cause — The indicated element is part of a complex element and is on a different level than its parent element.

Effect — Depending on what levels are being displayed, part of the element may be visible while other parts are invisible.

Effect — Some processes may ignore the fact that parts of the element are on different levels and treat the entire element as though it were on the level indicated by the header.

Solution — Use *FileFixer's* "Automatic Design File Repair" command with the repair setting *Settings* | *Change Settings* | *Repair...* | *Complex* | *Move Components To Same Level as complex header* turned on to move the component to the same level as its parent. This setting is OFF by default.

341 — A component is a different graphic group than its header.

Cause — Graphical component elements must have the same graphic group number as the complex header.

Effect – Unknown.

Solution — Select *Settings* | *Change Settings* | *Repair...* | *Complex* | *Set graphic group of component to match its header* to instruct *FileFixer* to change the graphic group number of the component element to match the header. (This setting is *not* on by default.)

343 — Reference file <_____> is missing.

Cause — The reference file may have been deleted.

Cause — The reference file may have been moved.

Effect — The missing reference file will not be displayed.

Effect — The missing reference file attachment will not plot.

Solution — Use *FileFixer's* "Automatic Design File Repair" command with the *Settings* | *Change Settings* | *Repair...* | *Reference File* | *Delete missing reference file attachment* setting turned on. This setting is OFF by default.

Solution — Correct the path of the reference file attachment using Axiom's *RefManager*. *RefManager* is particularly handy if error 343 is reported for a large number of your files.

Note: *FileFixer* searches `MS_RFDIR` and any configuration variable which is prepended to a reference file name (such as `MYPROJECT:MyFile.dgn`). For best results, ensure the MicroStation workspace of the workstation where you are running *FileFixer* matches the workspace of the project which created the file.

365 — Size of linkage is not a multiple of four.

Cause — The user linkage mentioned is not a multiple of four words in length. Note that this error is different from error 301 in that this error refers to the size of a *particular* attribute linkage (possibly one of many linkages) as determined by analyzing the size (in words) of that linkage itself (not the elementSize). Error 301 refers to the size of the entire linkage area. The linkage area is that part of the element after the graphic part of the element. The attribute area of an element may contain several individual attribute linkages.

Effect — The same effects that can be caused by error 301 can be caused by error 365.

Solution — Enter 365 into the string of error numbers in the text field for the *Settings* | *Change Settings* | *Repair...*| *Elements* | *Delete problem elements based on the error they generate* repair setting, and then use *FileFixer*'s "Automatic Design File Repair" command to delete elements which generate this error.

Solution — Use *Problem Element Viewer* to locate this element. Then redraw or delete the element.

373 —Vertex is off the edge of the design plane.

Cause — A line string's vertex is off the edge of the design plane.

Effect — MicroStation will not display this vertex.

Solution — Use *FileFixer's Settings* | *Change Settings* | *Repair...* | *Vertices* | *Fix line string's vertex off the edge of the design plane by:* repair option to either adjust the corrupted vertex or delete the element.

380 — Illegal element type detected.

Cause — An element type, which should never exist (type 0, 20, and 127), was detected.

Effect — The element will probably be ignored. It is illegal for such elements to exist and is probably an indicator of some other type of corruption. There is an excellent chance that there is an earlier elementSize error in the design file, which could make the design file undisplayable.

Solution — Use *FileFixer's* "Automatic Design File Repair" command with the *Settings* | *Change Settings* | *Repair...* | *Element* | *Remove illegal element types* setting (which is on by default). Be alert for other types of corruption in the vicinity of this element.

383 — Cell in a design file is nested within itself.

Cause — This element is a cell that is nested within a cell of the same name.

Effect — The cell may not display as expected.

Solution — Use the *FileFixer Settings* | *Change Settings* | *Repair...* | *Cell* | *Delete cell nested within itself* repair setting to delete occurrences of a cell that is nested within a cell of the same name. This setting is OFF by default.

V8-specific Warnings

V8-specific warnings are reported with numbers in the range 400–499.

For more information about structured storage, see the section titled “V8-specific Informational Messages.

410 — Missing Manifest data.

Cause — The file is missing one of the basic structured storage components.

Effect — Manifest data is missing for this file. This is suspicious and may be a symptom of file damage.

Solution — None needed at this time.

411 — Missing Session data.

Cause — The file is missing one of the basic structured storage components.

Effect — Session data is missing for this file. This is suspicious and may be a symptom of file damage.

Solution — None needed at this time.

412 — Missing Summary Information data.

Cause — The file is missing one of the basic structured storage components which contains the following file properties:

Author

Comments

Keywords

Last Saved By.

MicroStation version

Number

Revision

Subject

Title

Total Editing Time

Effect — Summary Information is missing for this file.

Solution — If you access design file properties from Windows or within MicroStation, you will need to create or *re*create the file properties for this file.

413 — Missing Document Summary Information data.

Cause — The file is missing one of the basic structured storage components which contains the following file properties:

Category (Windows) or Client (V8)

Company name

Manager

plus any *custom* properties which might be defined.

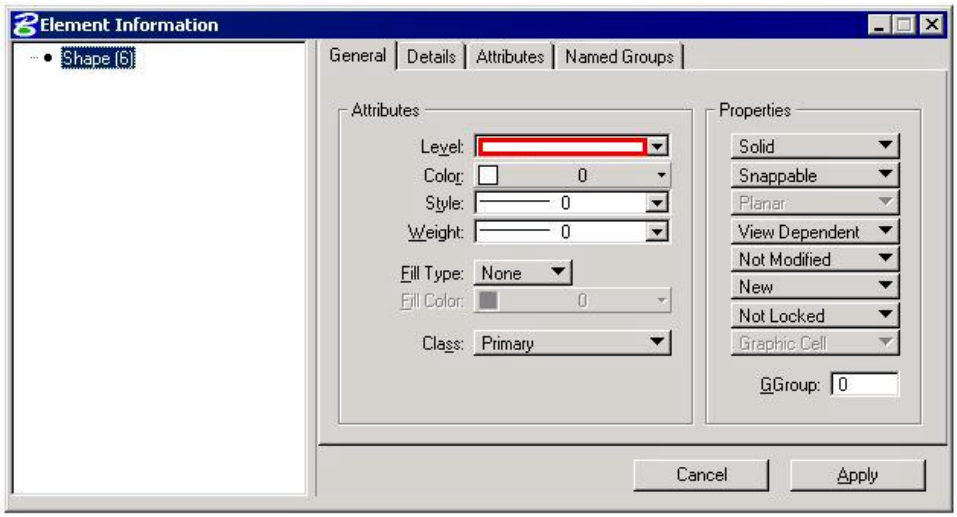
Effect — Document Summary Information is missing for this file.

Solution — If you access design file properties from Windows or within MicroStation, you will need to create or *re*create the file properties for this file.

415 — Bad element level.

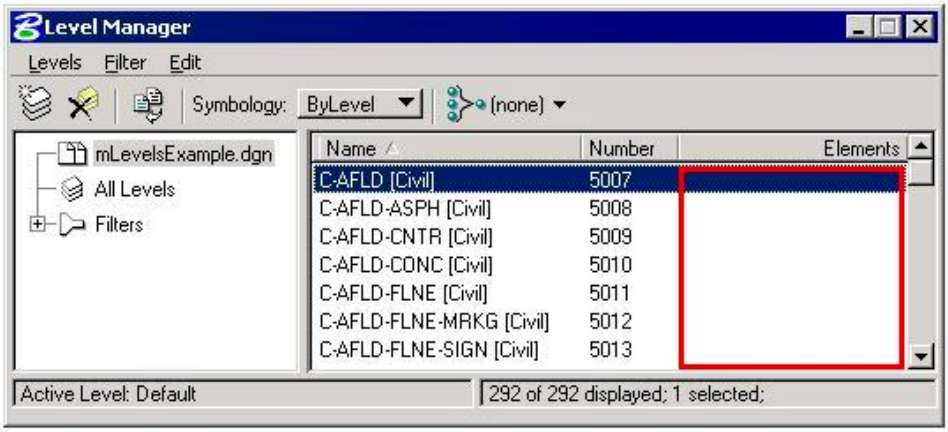
Cause — Element contains an undefined level (not present in the level table).

Effect — MicroStation's "Analyze" command cannot display the element's level.



Effect — Level does not display in MicroStation's Level Manager.

Effect — Levels appear to be empty (no elements) in MicroStation's Level Manager.



Solution — Use *FileFixer's "Automatic Design File Repair"* command *Settings* | *Change Settings* | *Repair...* | *Level* | *Repair elements on corrupted levels* to correct this problem. With this repair setting you can move all elements to the same level, move all elements to the default level (level number 0) or move each element to a new, automatically created level. For component elements you also have the choice of moving them to the same level as its complex header element. The default setting is to move each element to a new, automatically created level.

420 — Reference file attachment (_____) using slot zero found.

Cause — A reference file is in slot zero.

Effect — Some applications, including ProjectWise, do not recognize an attachment in slot zero. ProjectWise for example will not copy out this reference file when the master file is checked out.

Solution — Use *FileFixer's "Automatic Design File Repair"* command *Settings* | *Change Settings* | *Repair...* | *Reference File* | *Move slot zero reference to* | *The first available slot* or *The slot after the highest numbered slot currently used* to correct this problem.

429 — Multiple Master File Level Numbers.

Cause — Two or more master file level table level numbers are the same.

Effect — Under MicroStation XM only the first level with a duplicated level number is used. This can cause element's to display incorrectly when the element's symbology is based on the level's symbology.

Solution — First use *FileFixer's* "Delete the duplicate level number by:' by either keeping the first or last instance of the duplicate level number. Then recreate the deleted level if needed. *FileFixer's* repair setting is ON by default and keeps the first instance of the duplicate level number.

437 — Element's and model's are different.

Cause — An element's dimension is different than its model's dimension. For example, a 3D line in a 2D model.

Effect — Cannot modify the element.

Solution — Use *FileFixer's Settings* | *Change Settings* | *Repair...* | *Element* | *Set element's dimension to match the model's dimension.* This setting is OFF by default.

443 — Reference file is hidden.

Cause — The reference file attachment is hidden.

Effect — Reference file is not displayed in MicroStation's "References" dialog.

Solution — Use *FileFixer's Settings* | *Change Settings* | *Repair...* | *Reference File* | *Display hidden reference files* option to display hidden reference file attachments. This setting is OFF by default.

444 — Level is hidden.

Cause — The level is hidden.

Effect — The level is not displayed in MicroStation's Level Manager.

Solution — Use *FileFixer's Settings* | *Change Settings* | *Repair...* | *Levels* | *Display hidden levels* option to display hidden levels in MicroStation's Level Manager. This setting is OFF by default.

450 — Project Explorer stream count is corrupted.

Cause — The Project Explorer stream count is significantly larger than expected.

Effect — The design file may take an extraordinarily long time to open.

Solution — Use *FileFixer's Settings* | *Change Settings* | *Repair...* | *Structured Storage* | *Repair Project Explorer data* option to fix this problem.

Future Problems — File is in Danger

Future problems are reported with error numbers in the range 500–599. Continued use of the file will probably eventually cause severe problems.

514 — Attribute offset error.

Cause — The element's attribute offset was not what *FileFixer* expected. This may indicate the element's linkage area is damaged.

Effect — Depends upon the type of linkage. If the element already has a database linkage, related database information may be inaccessible due to this error. Trying to add a database linkage to this element may cause file corruption.

Solution — Use *Problem Element Viewer* to locate this element so that you can recreate the element and its database link, or delete it.

544 — New line flag missing from a text element linkage in a text node.

Cause — New line information is missing from text formatting linkage in the text element in a text node.

Effect — If you edit such text before fixing this corruption with *FileFixer*, all the text elements in the text node will be collapsed into a single line of text. This corruption must be prevented — it cannot be repaired once the corrupt text has been edited.

Solution — Use *FileFixer's* "Automatic Design File Repair" command with the *Settings | Change Settings | Repair... | Linkage | Add new line flag missing in a text element linkage that is part of a text node* option turned on (it is OFF by default) to set the line feed flag in the text formatting linkage.

545 — Duplicate text formatting linkage in a text node.

Cause — More that one text formatting linkage on the text elements in a text node.

Effect — If you edit such text before fixing this corruption with *FileFixer*, all the text elements in the text node will be collapsed into a single line of text. This corruption must be prevented — it cannot be repaired once the corrupt text has been edited.

Solution — Use *FileFixer's* "Automatic Design File Repair" command with the *Settings | Change Settings | Repair... | Linkage | Delete duplicate text formatting linkage in a text node* option turned on (it is OFF by default) to delete the duplicate text formatting linkages and to set the line feed flag in the remaining text formatting linkage.

560 — Text string overflows the element's length.

Cause — The number of characters is incorrect. It says there are more characters than actually exist.

Effect — The design file may become corrupted. A phantom element may get created.

Solution — Use *FileFixer* to correct the number of characters field in text element. The repair setting that corrects this error is *Settings | Change Settings | Repair... | Text | Correct the number of characters field in text element* and is on by default.

570 — Null cell name in cell library or design file.

Cause — There is an unnamed type 2 cell in a cell library.

Unnamed cells are not the same as orphan cells. Both have no name, but "orphan" cells have their H-bit set.

Nameless cells should not exist in a cell library unless their H-bit is set (which makes it an "orphan cell").

Solution — Use *FileFixer* to change the nameless cell to an orphan cell by setting its H-bit. The repair setting is *Settings* | *Change Settings* | *Repair...* | *Cell* | *Correct null cell names (type 2s only) by setting the H-bit.* This setting is on by default.

Serious V8-specific Errors

Serious V8-specific errors are reported with numbers in the range 600–699.

For more information about structured storage, see the section titled “V8-specific Informational Messages”.

601 — Missing “Default” model storage.

Cause — The default model is missing. [The Dgn-Md#000000 storage is missing.]

Effect — MicroStation may not be able to open the file.

Solution — Use *FileFixer’s* “Automatic Design File Repair” command with the *Settings* | *Change Settings* | *Repair...* | *Structured Storage* | *Repair severely damaged internal file structure* turned on (OFF by default).

Send your file as email attachment to support@AxiomInt.com if file size is less than one megabyte. Upload larger files to ftp://AxiomInt.com/incoming and notify support@AxiomInt.com by e-mail.

602 — Missing non-model storage.

Cause — The file is missing one of the basic structured storage components. [The Dgn^Nm storage is missing.]

Effect — MicroStation cannot open the file.

Solution — Send your file as email attachment to support@AxiomInt.com if file size is less than one megabyte. Upload larger files to ftp://AxiomInt.com/incoming and notify support@AxiomInt.com by e-mail.

603 — Missing non-model data.

Cause — The file is missing one of the basic structured storage components. [The Dgn^Nm$1 stream is missing.]

Effect — MicroStation cannot open the file.

Solution — See error #601.

610 — Missing models storage.

Cause — The file is missing one of the basic structured storage components. [The Dgn-Md storage is missing.]

Effect — MicroStation cannot open the file.

Solution — See error #601.

619 — Element is beyond the edge of the design plane.

Cause — According to its range block, at least one of this element's coordinates is *beyond* the edge of the design plane (or design cube).

This is possible because an element's IEEE floating point data can hold erroneous values which exceed the maximum allowed V8 design file coordinate.

Effect — Element may be unselectable, unmovable or excluded from fence operations. You may experience anomalies with actions such as plotting and Save As.

Solution — Use *FileFixer's* "Automatic Design File Repair" command to adjust element coordinates back inside the design plane (or design cube) boundary.

Solution — Use *Problem Element Viewer* to see what this element looks like and where it is located. If this is a stray, unwanted element and is actually causing plotting (or other) problems, then use *Problem Element Viewer* to delete it.

Solution — Add "619" to the "Delete problem elements based on the error they generate" listed under *Settings* | *Change Settings* | *Repair...* | *Element* to have *FileFixer* delete such elements in "Automatic Design File Repair" mode.

620 — Missing Header data.

Cause — The file is missing one of the basic structured storage components. [The Dgn~H stream is missing.]

Effect — MicroStation cannot open the file.

Solution — See error #601.

630 — Multiple master file level tables.

Cause — Design file contains more than one master file level table.

Only one of the master file level tables will be used. Prior to MicroStation V8.5.2.35 the *first* master file level table was always used. Starting with MicroStation V8.5.2.35 the master file level table with the lowest element ID is used.

Effect — Sometimes elements display with the correct level settings and other times the elements display with incorrect level settings.

Effect — Master file levels settings appear to match the level definitions for a reference file attachment.

Effect — MicroStation's Element Information tool displays elements with "blank" level names.

Effect — Message Center lists the error message "Detected multiple level tables in file…".

Solution — Use *FileFixer's* "Automatic Design File Repair" command with the *Settings* | *Change Settings* | *Repair...* | *Levels* | *Delete unused master file level tables* option turned

on (it is OFF by default) to delete unused master file level tables. You must choose which master file level table to retain (by default, the master file level table with the lowest element ID is retained).

Serious Errors — Problems are Likely

Serious errors are reported with error numbers in the range 700–799. There is a significant possibility that the elements involved will manifest problems of some kind.

702 — Phantom element. Complex bit is set for a non-component element.

Cause — An element that is not a component of a complex element has its complex bit set.

Effect — The element may not be selectable.

Solution — Use *FileFixer's* "Automatic Design File Repair" command with the *Settings* | *Change Settings* | *Repair...* | *Complex* | *Correct complex bit errors* turned on to fix this problem automatically. This setting is on by default.

703 — Complex bit is not set for a component element.

Cause — An element that is a component of a complex element does not have its complex bit set.

Effect — When component element complex bits are not properly set, problems can occur when a complex element is deleted.

Solution — Use *FileFixer's* "Automatic Design File Repair" command with the *Settings* | *Change Settings* | *Repair...* | *Complex* | *Correct complex bit errors* turned on to fix this problem automatically. This setting is on by default.

704 — Low range is greater than high range.

Cause — The x-low value is greater than the x-high value, the y-low value is greater than the y-high value, or the z-low value is greater than the z-high value. The z-range is checked for 3D models only.

Cause — This error is also reported when range's z-low or z-high is non-zero for an element in a 2D model.

Effect — Such elements may be ignored by plotting software and fence contents commands.

Solution — Use *FileFixer's* "Automatic Design File Repair" command with *Settings* | *Change Settings* | *Repair...* | *Range* | *Repair range of displayable elements* setting turned on. This setting is on by default. *FileFixer* will automatically swap the low and high values.

Solution — Use *Problem Element Viewer* to locate the element graphically. Delete or redraw.

Note: *FileFixer* cannot repair a 704 error for a complex header element which does not have any components — when there are no graphical components, the graphical range cannot be calculated.

705 — Element range doesn't match the min-max of its vertices.

Cause — The range of an element with vertices, points, poles, etc. is incorrect. The element range and the min-max of the vertices, points, poles, etc. that make up the element don't fall within the user-specified tolerance. This error can apply to lines, line strings, shapes, curves, conics, b-spline pole elements and pole strings.

Effect — This element may be improperly ignored by fence contents manipulation commands.

Effect — It may be impossible to snap to or select this element for manipulation.

Solution — Use *FileFixer's* "Automatic Design File Repair" command with *Settings* | *Change Settings* | *Repair...* | *Range* | *Repair range of displayable elements* setting turned on. This setting is on by default.

707 — B-spline pole element range doesn't match the min-max of its poles or the range of its b-spline surface header.

Cause — The range of a b-spline pole element is incorrect. MicroStation normally sets the range of this element type to the range of its type 24 b-spline surface header. The range of this element doesn't match.

Effect — This element may be improperly ignored by fence contents manipulation commands.

Effect — It may be impossible to snap to or select this element for manipulation.

Solution — Use *FileFixer's* "Automatic Design File Repair" command with *Settings* | *Change Settings* | *Repair...* | *Range* | *Repair range of displayable elements* setting turned on. This setting is on by default.

709 — Type __ header misses type __ component by __%.

Cause — All type 25 and type 21 elements that are components of b-spline curves should have the exact same range as their parent header element. The range of this element didn't match its parent.

Effect — Possible element select, snapping, and fence manipulation problems.

Solution — Use *FileFixer's* "Automatic Design File Repair" command to fix this type of problem automatically.

712 — Range of right circular truncated cone is incorrect.

Cause — *FileFixer* calculates the range of right circular truncated cones based on the locations of their centers, radii, and quaternions. The range calculated by *FileFixer* for this element doesn't match the range stored in its element header.

Effect — Possible element snapping, selection, and fence manipulation problems.

Solution — Use *FileFixer*'s "Automatic Design File Repair" with the *Settings* | *Change Settings* | *Repair...* | *Repair range of displayable elements* setting on to repair this element. This setting is on by default.

717 — Cell class map is incorrect.

Cause — This problem occurs when the list of classes (stored in the cell header) used by elements in the cell does not match the actual classes used by elements in the cell.

Solution — Use *FileFixer's* "Automatic Design File Repair" command with the *Settings* | *Change Settings* | *Repair...* | *Element* | *Set illegal class to:* repair setting turned on to correct this problem automatically. This setting is OFF by default.

Note: Cell component classes are stored in the cell's *class map*. In the following examples, bits are counted right to left beginning with 0:

0000 0000 0000 0000 EMPTY (or NULL) — no graphical cell components.*

0000 0000 0000 0001 class 0 — cell contains a primary class element.

0000 0000 0000 0100 class 2 — cell contains a construction class element.

0000 0000 0000 0101 classes 0 and 2 (primary and construction) — cell contains primary and construction class elements.

* A cell's class map can be EMPTY when the cell is composed of nested cells or other non-graphic elements. An element's class is stored in its *display header* which only exists in graphical elements (only *graphical* elements have a class).

720 — Complex header range is off by ___%.

Cause — The union of the range blocks of the components of the complex element header does not match the range block of the complex header element itself. The percentage is determined by dividing the deviation from perfection as calculated by *FileFixer* (in UORs) by the size of the element (in UORs). This test is performed on the x, y, and (for 3D models) z axes. The worst (largest) of the error percentages is the one reported. Note that if the size of the element (along the z-axis as an example) is very small (lets say 1 UOR as an example), then a relatively small deviation from the ideal Z-High or Z-Low (lets say 50 UORs) would result in a large percentage deviation (for example, 5000%). Don't worry — just use the "Automatic Design File Repair" command to adjust the problem elements.

Effect — The complex element may be improperly excluded from fence manipulation and plotting commands.

Solution — Use *FileFixer*'s "Automatic Design File Repair" with the *Settings* | *Change Settings* | *Repair...* | *Repair range of complex header elements* setting on to repair this element. This setting is on by default.

721 — An element of this type cannot be a component of a complex header of the specified type.

Cause — Each type of complex element (like cells and text nodes) can only contain certain types of elements. For example, a text node (type 7) can contain only text elements (type 17s). A connected string (type 12) can contain only linear elements such as lines (type 3s), line strings (type 4s), arcs (type 16s), etc. The element just read in cannot legally be contained in the complex element header mentioned near the end of the message.

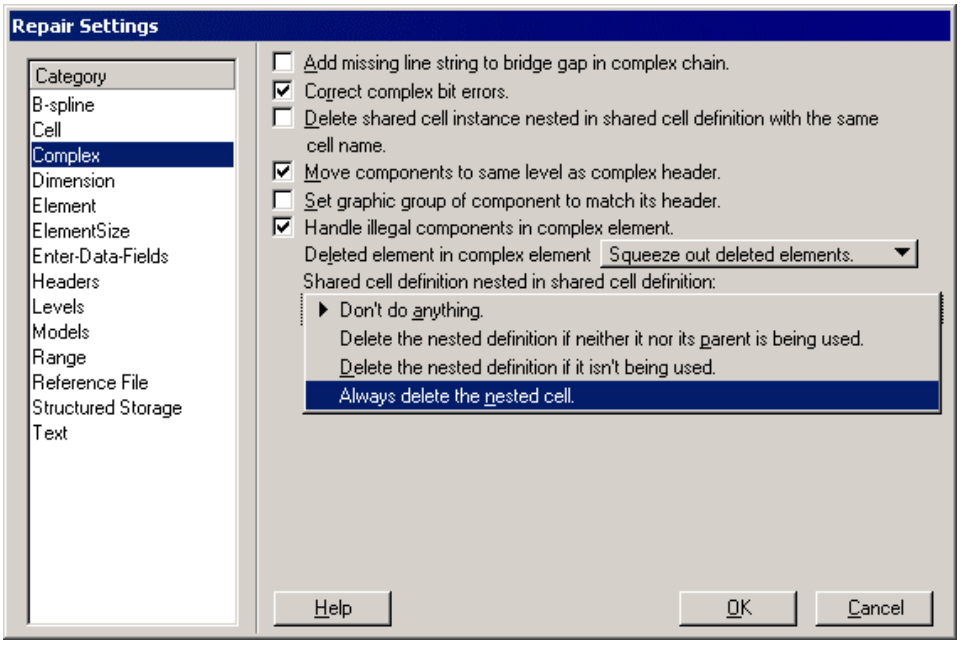
Cause — The complex header element mentioned contains a words-in-description error that causes it to think it contains elements that really should be after its end.

Effect — Some combinations of elements can be unselectable.

Solution — Check *Settings* | *Change Settings* |*Repair* | *Complex* | *Handle illegal components in complex element* for settings to handle this error depending on the type of situation reported.

Solution — For problems with complex graphic elements, you can also use *Problem Element Viewer* to visually inspect both the complex element as a whole and the unwanted component. Redraw or delete.

Solution — Activate the following *FileFixer* Repair setting to clean up nested shared cell definitions:



722 — Dimension element with no data points or associative points.

Cause — A dimension element must have at least one data or associative point.

Effect — Element may be unselectable.

Solution — Use *FileFixer's Settings* | *Change Settings* | *Repair...* | *Dimension* | *Remove dimension element with no data points or associative points* repair setting to delete the bad dimension element. This setting is on by default. You will then need to recreate the dimension element.

724 — Range of arc or ellipse is incorrect.

Cause — *FileFixer* calculates the range of arcs and ellipses based on the length of their axes, angle of rotation, origin, start angle and sweep angle. The range calculated by *FileFixer* for this element doesn't match the range stored in its element header.

Effect — Possible element snapping, selection, and fence manipulation problems.

Solution — Use *FileFixer*'s "Automatic Design File Repair" with the *Settings* | *Change Settings* | *Repair...* | *Repair range of displayable elements* setting on to repair this element. This setting is on by default.

725 — Range of text element is incorrect.

Cause — The range of this text element is incorrect.

Cause — *FileFixer* calculates the range of 2D and 3D text elements based on their origin, number of characters, rotation or quaternions, length multiplier, and height multiplier.

Effect — Possible element snapping, selection, and fence manipulation problems.

Solution — Use *FileFixer*'s "Automatic Design File Repair" with the *Settings* | *Change Settings* | *Repair...* | *Repair range of displayable elements* setting on to repair this element. This setting is on by default.

Be sure you have the correct font library specified in MicroStation.

727 — Range of text node element is incorrect.

Cause — *FileFixer* has calculated that the range of an empty text node is incorrect.

Effect — Possible element snapping, selection, and fence manipulation problems.

Solution — Use *FileFixer*'s "Automatic Design File Repair" with the *Settings* | *Change Settings* | *Repair...* | *Repair range of displayable elements* setting on to repair this element. This setting is on by default.

733 — A shared cell instance is nested in a shared cell definition with the same name.

Cause — A shared cell instance cannot be nested in a shared cell definition that defines that shared cell instance.

Effect — MicroStation cannot open this file.

Solution — Use *FileFixer*'s "Automatic Design File Repair" with the *Settings* | *Change Settings* | *Repair...* | *Delete shared cell instance nested in a shared cell definition with the same name* setting on to repair this element. This setting is on by default.

750 — Severe elementSize errors.

Cause —The element's size ("elementSize" property) is not correct.

Effect — This element and subsequent elements will not display in MicroStation. [The exact number of elements which will not display is dependent upon element distribution within the structured storage file.]

Solution — Restore from backup if possible.

Send your file as email attachment to support@AxiomInt.com if file size is less than one megabyte. Upload larger files to ftp://AxiomInt.com/incoming and notify support@AxiomInt.com by e-mail.

Note: There is no search setting for this error — due to the severity of the condition it is always reported when found.

764 — Siamese attribute linkage (linkage overlaying the next element).

Cause — The element's attribute linkage extends beyond the end of the element. That means the linkage overlaps the *next* element in the file.

Effect — Applications that depend upon data in the linkage area could malfunction or crash. Accessing a database using this linkage could fail.

Effect – Element may not be able to be moved without MicroStation V8 locking up or aborting.

Effect – Design file or model cannot be plotted without MicroStation V8 locking up or aborting.

Solution — Use *FileFixer's* "Automatic Design File Repair" command with the *Settings* | *Change Settings* | *Repair...* | *Linkage* | *Remove siamese linkage* setting turned on. This setting is off by default.

767 — Shared cell instance dependency link to its shared cell definition is broken.

Cause — The dependency linkage on the shared cell instance contains an element ID that does not match any shared cell definition in the design file.

Effect — This cell will not display even if you add a shared cell definition with the same name.

Solution — Use *FileFixer's* "Automatic Design File Repair" command with the *Settings* | *Change Settings* | *Repair...* | *Linkage* | *Set the shared cell instance dependency linkage's element ID to the shared cell definition's element ID.*

768 — Shared cell instance has no matching shared cell definition.

Cause — Whenever there is a shared cell instance (type 35) in a design file, there must also be a shared cell definition element (type 34) with the same name in the design file. The type 34 contains the instructions on how to display the type 35. Error number 768 indicates a type 35 has no corresponding type 34 — the type 35 (instance) cannot display (no definition). The shared cell definition element must have the *exact* same name as the shared cell instance.

Effect — This cell will not display (no definition).

Solution — Find (or create) a cell library that contains a cell of the specified name. Attach that cell library. Place that cell as a shared cell (this should insert a shared cell definition into your file). Delete the *instance* you just placed (not needed). Do a FIT. The original cell (and all other shared cell instances with the same name) should now be visible. Use Axiom product *Problem Element Viewer* to zoom in on the repaired cell if you don't know where it is located.

Solution — *FileFixer's* "Automatic Design File Repair" command with the *Settings* | *Change Settings* | *Repair...* | *Cell* | *Create dummy shared cell definition for missing definition* setting turned on (OFF by default). This setting will create a dummy shared cell definition. What this setting does is create a shared cell definition element with a text element as its only component. The text element contains the name of the shared cell. This will graphically show you the locations where the cell is used. Use Axiom product *Problem Element Viewer* to zoom in on the repaired cell if you don't know where it is located.

Solution — If the above solutions do not cause the specified cell to become visible, or if the cell displays incorrectly, you should use *Problem Element Viewer* to delete the type 34 and type 35 and use MicroStation to replace the cell from the cell library. Even if the cell is displaying incorrectly, *Problem Element Viewer* may be able to at least show where in the design file, the problem cell is located.

787 — Element has too many vertices.

Cause — Elements like line strings, shapes and curves can have no more than 5000 vertices. This element has a number of vertices value greater than 5000.

Effect — Element may not display and view manipulations can cause MicroStation to abort.

Solution — Select the *Settings* | *Change Settings* | *Repair...* | *Vertices*| *Set number of vertices to maximum allowed* setting to automatically adjust the number of vertices to the maximum allowed of 5000.

788 — B-spline pole element (__) has too few poles.

Cause — This type 21 element has less than two poles.

Effect — Element may not display as expected.

Solution — Use *Problem Element Viewer* to delete the parent complex header of this element. The element number of the parent is shown in parentheses in *FileFixer's* report.

789 — Element has too few vertices, points, boundaries, etc.

Cause — Elements with vertices, points, etc. must have a minimum number of these. This element did not have enough.

Effect — Element may not display as expected. You may experience plotting anomalies.

Solution — Select the *Settings* | *Change Settings* | *Repair...* | *Element* | *Remove elements with too few vertices* setting to automatically remove these elements from the design file. This setting is on by default.

Solution — Use Axiom's Problem Element Viewer to determine if the erroneous element is displayable. If the element is displayable, you may want to redraw the element before deleting it.

Fatal V8-specific Errors

Fatal V8-specific errors are reported with numbers in the range 800–899.

For more information about structured storage, see the section "V8 specific Information Messages".

For more information about structured storage repair, see the section "Structured Storage".

801 — Missing header data for model <_____>.

Cause — Models consist of a header (contains model's name, description and other information), non-graphical elements (like reference file attachments) and graphical elements. This error indicates the model's header information is missing. [The ~Mh stream is missing for the indicated model.]

Effect — The model cannot be selected from MicroStation's Models dialog. MicroStation may not be able to open the file.

Solution — Select *FileFixer*'s *Settings* | *Change Settings* | *Repair...* | *Structured Storage* | *Rebuild damaged model header data* setting (on by default). *FileFixer* uses Design History, if present, to reconstruct the model header. If Design History is not available, *FileFixer* reconstructs the model header using the model index stream.

Note: If Design History is not available, *FileFixer* cannot recover the model's original working units and working unit labels. Working units can be easily re-set using MicroStation (*Settings* | *Design file* | *Working units*). *FileFixer's* report will remind you to do so when necessary.

802 — Data exists but cannot be read.

Cause — This error indicates structured storage data is present, but is damaged and cannot be read. Specifically, an entire sub-file ("stream") cannot be accessed. The type and location of the damaged data (stream) is indicated in *FileFixer's* report. This error is much more severe than a corrupted individual element — this error indicates a large

portion of the file (an entire stream), possibly containing hundreds of elements, cannot be read.

Effect — The manifestations you experience will depend on what data is damaged.

Stream Type	Manifestation when damaged
Model header	If the active model (last saved) is damaged, you will likely encounter a MicroStation exception when you attempt to open the file. If the damaged model is not the active model, the model cannot be selected from MicroStation's Models dialog.
Graphical	None of the elements in the stream (possibly hundreds) will display.
Non-graphical	Reference files will not display on screen or in References dialog.
Non-Model	The file's level table, line styles, shared cells and other global data, or a portion thereof, will be lost and unusable.

Examples of error 802 that might appear in your *FileFixer* report:

```
*** -- Non-model data exists but cannot be read. [Dgn^Nm\$2 stream cannot be
read.]  [802].

*** -- Graphical data exists but cannot be read. [131 elements cannot be
extracted from Dgn-Md\#000000\Dgn^G\$3 stream.]

*** -- Non-graphical data exists but cannot be read. [3 elements cannot be
extracted from Dgn-Md\#000000\Dgn^C\$1 stream.]  [802].

*** -- Graphical data exists but cannot be read. [Dgn-Md\#000000\Dgn~Mh stream
cannot be read.]  [802].
```

Note: Error 802 is reported for *any* damaged stream found. Since all data in a V8 file is stored in one stream or another, error 802 encompasses *all* possible data which can be stored in a V8 design file. Some design file streams, when damaged, impact users and MicroStation more severely than other damaged streams. From analysis of customer files, we found the most common and most catastrophic stream category to be damaged are model header streams. Accordingly, we implemented automated repair of model headers first.

Solution — If *FileFixer* reports that a model header is damaged, select the *Settings* | *Change Settings* | *Repair...* | *Structured Storage* | *Rebuild missing or damaged model header data* setting (on by default). *FileFixer* uses Design History, if present, to reconstruct the model header. If Design History is not available, *FileFixer* reconstructs the model header using the model index stream. See description of error 801 for more information about model header repair.

Solution — If *FileFixer* reports that a non-model stream is damaged, find a healthy file (or dgnlib) from the same project and enter:

```
levelmanager library attach <name of healthy file>

dgnlib update all
```

This will refresh your level table, dimension styles, text styles and multi-line styles.

Solution — If *FileFixer* reports that a non-model stream is damaged, select the *Settings* | *Change Settings* | *Repair...* | *Structured Storage* | *Recover elements beyond missing or damaged non-model data* setting (OFF by default).

Solution — If *FileFixer* reports that a model's non-graphical stream is damaged, select the *Settings* | *Change Settings* | *Repair...* | *Structured Storage* | *Recover elements beyond missing or damaged non-graphical data* setting (OFF by default).

Solutions — If *FileFixer* reports that a model's graphical stream is damaged, restore from DgnHistory or backup if available.

Solutions — If *FileFixer* reports that a model's graphical stream is damaged, select the *Settings* | *Change Settings* | *Repair...* | *Structured Storage* | *Recover elements beyond missing or damaged graphical data* setting (OFF by default).

Note: You can always contact Axiom support for assistance with file recovery: support@AxiomInt.com.

819 — Graphical element in non-graphical data.

Cause — Graphical element in the non-graphical data area. The non-graphical data area is for reference file attachments, application data elements, etc.

Effect — MicroStation V8i may lock up or abort.

Solution – Turn on *FileFixer's Settings* | *Change Settings* | *Repair...* | *Element* | *Delete graphical element in non-graphical data* setting to delete the graphical elements in the non-graphical area. This repair setting is OFF by default.

820 — The design file cannot be opened. Replacing the model headers.

Cause — Missing or severely damaged model header or model index

Effect — MicroStation V8 fails to open the design file.

Solution – Turn on *FileFixer's Settings* | *Change Settings* | *Repair...* | *Structured Storage* | *Rebuild missing or damaged model header or index data* to replace or repair the missing or damaged model header data.

850 — Project Explorer stream count is corrupted.

Cause — The Project Explorer stream count is significantly larger than expected.

Effect — The design file may take an extraordinarily long time to open or to repair.

Solution — Use *FileFixer's Settings* | *Change Settings* | *Repair...* | *Structured Storage* | *Repair Project Explorer data* option to fix this problem.

Note: When the Project Explorer data corruption is so severe that *FileFixer* would appear locked up when attempting to resolve it, *FileFixer* stops processing if the repair option, described in the solution section above, is not turned on. When this occurs *FileFixer* reports an 850 error. If the problem is relatively minor or the repair option is turned on *FileFixer* reports a 450 error and processing is not interrupted.

897 — Non-model section is corrupted.

Cause — The first non-model stream is corrupted.

Effect —MicroStation cannot open the file. MicroStation must find the design file header in order to open the file. This element is normally in the first non-model stream.

Solution – Turn on *FileFixer*'s *Settings* | *Change Settings* | *Repair...* | *Structured Storage* | *Repair severely damaged internal file structure* setting to repair the damaged V8 design file. This repair setting is OFF by default.

Solution — Send your file as email attachment to support@AxiomInt.com if file size is less than one megabyte. Upload larger files to ftp://AxiomInt.com/incoming and notify support@AxiomInt.com by e-mail.

898 — Cross-linked streams found.

Cause — Two different data streams (subfiles) believe their data occupies the same location in the file.

Effect — MicroStation cannot open the file.

Solution – Turn on *FileFixer*'s *Settings* | *Change Settings* | *Repair...* | *Structured Storage* | *Repair severely damaged internal file structure* setting to repair the damaged V8 design file. This repair setting is OFF by default.

Solution — Send your file as email attachment to support@AxiomInt.com if file size is less than one megabyte. Upload larger files to ftp://AxiomInt.com/incoming and notify support@AxiomInt.com by e-mail.

899 — Internal storage in this file is corrupted.

Cause — The structured storage framework of this file is severely damaged, preventing access to all data in the file.

Effect — MicroStation cannot open the file.

Solution – Turn on *FileFixer*'s *Settings* | *Change Settings* | *Repair...* | *Structured Storage* | *Repair severely damaged internal file structure* setting to repair the damaged V8 design file. This repair setting is OFF by default.

Solution — Send your file as email attachment to support@AxiomInt.com if file size is less than one megabyte. Upload larger files to ftp://AxiomInt.com/incoming and notify support@AxiomInt.com by e-mail.

Note: The file `templateV8.dgn` (delivered with *FileFixer for V8*) is required to repair error 899.

Fatal Errors

Fatal errors are reported with error numbers 900–999. It is unlikely that any MicroStation application will be able to process the file in its entirety without a severe error.

902 — First element of file is not an active type 9 (design file header).

Cause — This is a design file or cell library whose first element (type 9 design file header) has been deleted or is badly damaged.

Effect — MicroStation cannot open the file.

Solution — Select *FileFixer*'s *Settings* | *Change Settings* | *Repair...* | *Headers* | *Replace the design file header element*. This option will add the type 9 design file header as the first element of the design file. Since the design file header element is deleted or badly damaged you must also turn on the *Settings* | *Change Settings* | *Repair...* | *Structured Storage* | *Repair severely damaged internal file structure* option so this fatal corruption can be fixed before attempting to open the file using MicroStation functions in later *FileFixer* steps.

905 — Duplicate b-spline poles in a closed b-spline have been found.

Cause — A b-spline pole element (type 24) has duplicate first and/or last poles in a closed b-spline. MicroStation normally doesn't create b-spline poles in this manner. Converting files from DWG to DGN sometimes results in this.

Solution — Select *FileFixer*'s *Settings* | *Change Settings* | *Repair...* | *B-spline* | *Open the closed b-spline that has duplicate b-spline poles* setting to change the b-spline from closed to open. This repair setting is OFF by default.

911 — Design file header (type 9) should be on level 8, not _____.

Cause — The design file header's element is not on level number 8.

Effect — MicroStation V8 will not be able to open the design file. MicroStation V8 will display "Unable to open design file. Please contact your local site administrator or technical support provider for further assistance."

Solution — Use *FileFixer*'s "Automatic Design File Repair" mode with the *Settings* | *Change Settings* | *Repair...* | *Headers* | *Correct the design file header's level* option to repair this element. This setting is ON by default.

912 — Illegal floating point value encountered.

Cause — An arc or ellipse element contains a value that cannot be converted into a floating point number.

Solution — Use *Problem Element Viewer* to locate the element graphically so that you can delete and/or redraw.

Chapter 14 — Batch Mode

What is batch mode?

FileFixer can be run outside of MicroStation to process one or a *batch* of files.

You must have MicroStation installed, but MicroStation's graphical user interface is not used.

The details about running *FileFixer* in batch mode are covered in this section.

Important Note: Msbatch mode for *FileFixer* has been temporarily disabled under MicroStation V8i and later versions of MicroStation. This is necessary due to problems with several critical MDL functions that fail only in msbatch mode. You can still search and repair all your V8 design files launching *FileFixer* in the graphical user interface of MicroStation.

Why use batch mode?

Batch mode might be handy for you under any of the following conditions:

1. You want to schedule timed execution of *FileFixer*, especially after normal business hours.
2. You can't tie-up network resources during normal business hours and need *FileFixer* to run at a specified time, unattended.
3. Your server can't run MicroStation in graphics mode because it has no graphics terminals connected.

How do you run *FileFixer* in batch mode?

To run *FileFixer* in batch mode follow these steps:

1. To prepare for batch mode processing you must first run *FileFixer* within MicroStation by selecting *FileFixer* from the "Axiom" menu.
2. You can use *FileFixer's* defaults, but if you want to use customized settings, select the *FileFixer* search *and* repair settings you want to use in batch mode and save your settings with *Settings* | *Save Settings As...*.
3. Run MicroStation in batch mode. The syntax of the command you enter at the operating system command line is:

```
<path1>msbatch <path2>fixer.ma options design_file_wildcard
```

<path1> is the location of the file “MSBATCH.BAT”. By default, this path is “C:\Program Files\Bentley\Program\MicroStation\”, but this may be different on your computer. Check with your system administrator if you have any questions.

<path2> is the location of the file “FIXER.MA”. By default, this path is “C:\Program Files\Axiom\V8\Fixer\”, but this may be different on your computer. Check with your system administrator if you have any questions.

options are described below.

design_file_wildcard is any legal Windows path and filename or filename wildcard expression, such as “***.dgn**”.

For example, if MicroStation and *FileFixer* are installed in the default locations and your current directory is “C:\Program Files\Bentley\Program\MicroStation\”, the batch command to search for problems in `MyFile.dgn` would be:

```
msbatch "C:\Program Files\Axiom\V8\Fixer\fixer.ma" -sb
MyFile.dgn
```

If your current directory is a project directory containing design files, you need to specify the full path to msbatch.bat.

Note: Remember to put quotes around any pathname that has a space in it.

Valid options

-a — Automatic Design File Repair. Uses Automatic Design File Repair settings.

-b — Batch Mode. This option is *required* when running in batch mode.

-d — Reload design file. Reload the currently loaded design file once the files have been checked.

-f Report_File_Name — The name of the report file. Use this option when you don’t want to use the report file name you had last used when running *FileFixer* from within MicroStation.

-i List_File —In “Search for Problems” mode *FileFixer* can create a list of corrupted files found. *FileFixer* can save the names of *all* corrupted files, or you can exclude files with minor errors. You control the threshold of corruption severity by selecting *Settings | Search Settings | Design File | Severity of errors to be reported*. The name of the output file created is defined by the **List_File** option. The files written to **List_File** can be used as input in “Automatic Design File Repair” mode — on *FileFixer*’s main dialog click <Select> button and then select *File | Import File List...*.

-o Repaired_Design_File_Directory — Specify the directory where the repaired files will be placed.

-r Resource_File_Name — Specify which *FileFixer* resource file to use for the search and repair settings.

-s — Search Mode. Analyze, but do not repair files. Analysis report is created. Uses Search for Problems settings.

-u 999 — Only process those files which have been updated within the specified number of days — up to 999 days.

Examples

Note: For simplicity the following examples exclude paths for `msbatch.bat` and `fixer.ma`.

```
msbatch fixer.ma -ab MyFile.dgn

msbatch fixer.ma -sb "*.dgn, *.cel"

msbatch fixer.ma –sb -f MyOutput.log *.dgn

msbatch fixer.ma –ab -r
c:\progra~1\axiom\v8\fixer\fixer.ini *.dgn

msbatch fixer.ma –ab -o c:\progra~1\axiom\v8\fixer\output
*.dgn

msbatch fixer.ma –sb –i
c:\progra~1\axiom\v8\fixer\fixer.lst *.dgn

msbatch fixer.ma –sb –u 10 *.dgn
```

Tip: **design_file_wildcard** can also be the name of a file that contains the filenames you wish to process. For example, if you create a file called `MyFiles.txt` which contains one path and filename per line:

```
C:\MyProject\Floor.dgn

C:\MyProject\Walls.dgn

C:\MyProject\Ceiling.dgn
```

You can search for problems in all of the above files using the following command:

```
msbatch c:\progra~1\axiom\v8\fixer\fixer.ma –sb
@MyFiles.txt
```

Note: DOS commands are limited to 132 characters (on the command line).

Chapter 15 — Preventive Maintenance

Nightly *FileFixer* maintenance

Some problems, left unnoticed and unhandled, can get worse. Some design file problems can be prevented by fixing your files *before* the invalid data becomes obvious. *FileFixer's* 400 and 500 level errors are examples of such design file time bombs.

This is why some *FileFixer* users schedule *FileFixer* to repair all their design files on a regular basis.

By having *FileFixer* regularly repair your design files *before* they manifest fatal errors, you can prevent certain types of invalid data from becoming fatal.

Setting up a typical nightly *FileFixer* maintenance run

In this section, you will see how to set up *FileFixer* to run automatically in batch mode using the Windows scheduler.

By running *FileFixer* in "Search for Problems" mode, you can create a report on files that contain serious problems as well as a list of the problem files. *FileFixer* can later be used to repair only those files in the problem file list.

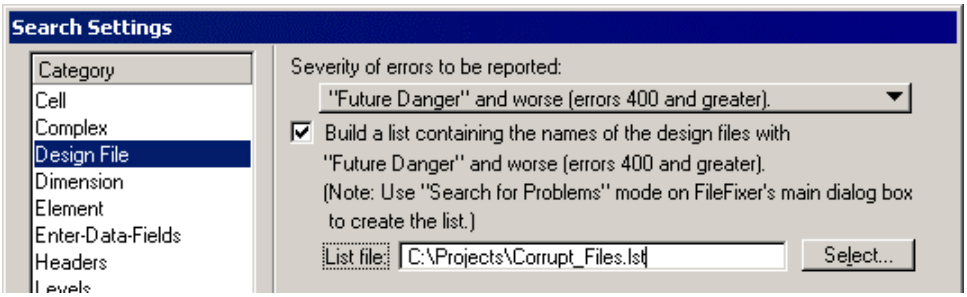
Scheduling

For best results, schedule *FileFixer before* file compression and *before* nightly backups.

Create the settings file for nightly *FileFixer* run

Start by opening *FileFixer* in graphic mode and set up the settings (repair and search) the way you want *FileFixer* to work.

1. You will probably want your nightly run to produce a list of corrupt files that *FileFixer* can later use to repair only those files that are seriously corrupted. Having *FileFixer* only repair those files which are seriously corrupted can save you processing time, especially valuable when you have large amounts of files to maintain. In the example below, we first selected “Future dangers and worse (error codes 400 and greater)”. This ensures that our list will only contain the names of files with errors of level 400 or higher. We also entered a name for the output list of corrupt files.



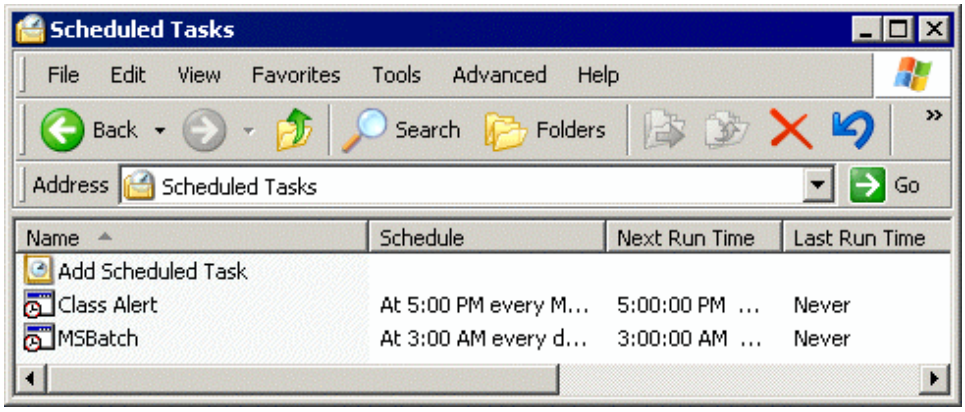
2. When you are done making your selections, press <OK> to return to the main *FileFixer* dialog box.

3. Save your settings using the *Settings* | *Save Settings As...* function from the main *FileFixer* dialog box. Give the settings file a unique name, “fixer_nightly.ini” for example, because you may only want to use these particular settings when running *FileFixer* at night.

Note: Those familiar with the Windows scheduler or using some other method can proceed to the next section to see how to set up the MSBATCH command.

Set up the Windows scheduler

1. Open the Windows scheduler by selecting *Start* | *Programs* | *Accessories* | *System Tools* | *Scheduled Tasks* in Windows. The following box should appear:


Scheduled Tasks
File Edit View Favorites Tools Advanced Help
Back Search Folders
Address Scheduled Tasks
Go
Name Schedule Next Run Time Last Run Time
Add Scheduled Task
Class Alert At 5:00 PM every M... 5:00:00 PM ... Never
MSBatch At 3:00 AM every d... 3:00:00 AM ... Never

2. If you do not have the “Scheduled Tasks” feature installed on your computer, please see your System Administrator about getting this feature installed.

3. Double-click “Add scheduled task”.

4. Click <Next> on the Scheduled Task Wizard box that appears.

5. In the next window, <Browse> to your MicroStation directory to find the MSBATCH.BAT file and select it.

6. Enter a name for the task in the next Wizard window (such as “*FileFixer*”), select “Daily” as in the image below and press <Next>.

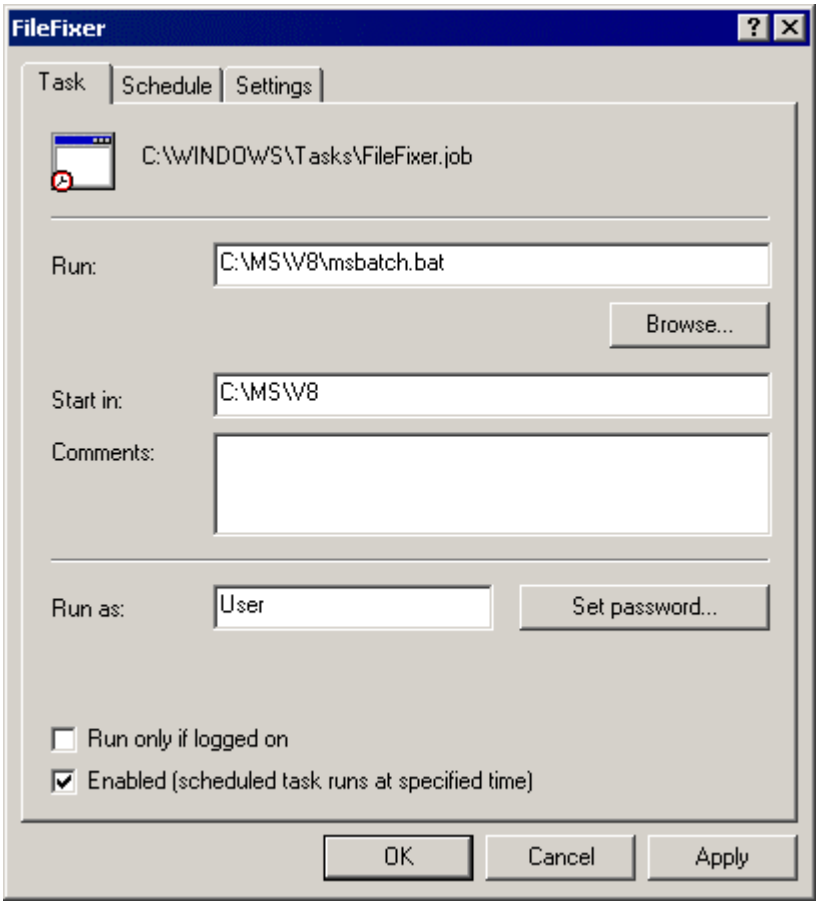
7. On the next screen you can select the time and day you want this task to start. Make your choices and press <Next>.

8. On the last screen, be sure to check the "Open advanced properties for this task when I click Finish" option. Then click <Finish>.

Scheduled Task Wizard
You have successfully scheduled the following task:
FileFixer
Windows will perform this task:
At 1:00 AM every day, starting 22 Aug 2007
Open advanced properties for this task when I click Finish.
Click Finish to add this task to your Windows schedule.
< Back
Finish
Cancel

9. A Properties box for this task will appear. This box will have whatever name you gave your task, such as "*FileFixer*".

Set up the *FileFixer* MSBATCH command



1. You will need four parameters in the “Run” box after “...\MSBATCH.BAT\”:

- The path to the *FileFixer* executable program.
- The code “-sb” tells *FileFixer* that the mode will be “Search for Problems” and tells msbatch to run it in batch mode.
- A path to the saved settings file.
- A path to the design files to be processed or to a list of files. Options for this path are described in detail in the section on “Batch Mode”.

Example:

```
C:\MS\V8\MSBATCH.BAT c:\ms\V8\axiom\fixer\fixer.ma -sb
C:\MS\V8\axiom\nightly.rsc c:\ms\V8\dgnfiles\*.dgn/s
```

2. When you click <OK> on this dialog box, your setup is complete.

Tip: For a detailed description of these fields and of other options available in batch mode, see “Batch Mode” section.

Chapter 16 — Tips

About yesterday's backup...

Probably the most common method of "repairing" design files in use in the world today is deleting the damaged file and reloading yesterday's backup.

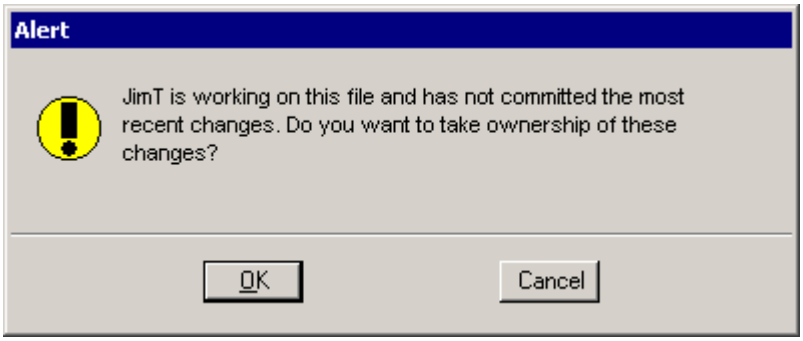
The most obvious problem with this "technique" is that you can lose up to a full day's worth of work. Considering the cost of a day's worth of operator and workstation time, clearly this should be a last resort.

But there is a more important reason for not going back to yesterday's backup. The error that caused your file to become corrupted today might be dormant in yesterday's backup. The safest solution is to analyze misbehaving or crashed design files with *FileFixer* and fully repair them.

Using *FileFixer* with Design File History

FileFixer's "Automatic Design File Repair" processing typically adds, modifies or deletes elements in a design file being processed. If you activated Design History for a file, *FileFixer's* changes will be recorded in the file's history.

If you attempt to use *FileFixer* to repair a file that has uncommitted changes created by another user, you will encounter a dialog box similar to the following:


Alert
JimT is working on this file and has not committed the most recent changes. Do you want to take ownership of these changes?
OK
Cancel

FileFixer's "Search for Problems" mode also generates a "Design file settings" *change* event when a seal-of-approval is created or updated. Creation of a seal-of-approval can be disabled by selecting *Settings* | *Change Settings* | *General...* | *Write seal-of-approval information to the design file.*

V7 files

FileFixer for V8 is designed to process V8 format design files and cell libraries.

If *FileFixer* fails to handle an error

When an element contains more than one error, *FileFixer* will report the error that it considers the most serious.

One error may be obscured by another error *FileFixer* considers more serious. Therefore, if *FileFixer* fails to fix a certain problem, try running *FileFixer* again on the *fixed* file. An error that was obscured the first time through may become fixable after more serious problems are corrected.

Controlling the types of errors which are reported

Sometimes you may not want *FileFixer* to report all the problems it finds.

For example, say you receive a design file from a contractor but he didn't send the font resource file he used. Well, if you don't have a font resource file with that font number, *FileFixer* will report the missing font for each text element in every file. Since you know about this problem (and will receive the font resource file from the contractor soon) you can turn off Search Setting *Text* | *Text element uses non-existent font* (error 144) so that *FileFixer* doesn't report the missing font. This will simplify the report output so you can immediately see the remaining problems that need to be addressed.

You can also control the severity of errors being reported, using *Settings* | *Change Settings...* | *Search* | *Design File* | *Severity of errors to be reported.*

Our commitment

You should never be satisfied with less than 100% perfect performance from this product. If you find any files that *FileFixer* cannot diagnose and/or repair or if *FileFixer* reports something that doesn't appear to be an error, send the file along with a description of the problem to us. We will do our best to find out what is wrong with your file and modify *FileFixer* to handle that problem in the future.

If you find so much as a comma out of place with this product or its documentation, we are interested.

Contact Axiom Support at support@AxiomInt.com to obtain instructions for uploading problem files to our ftp site.

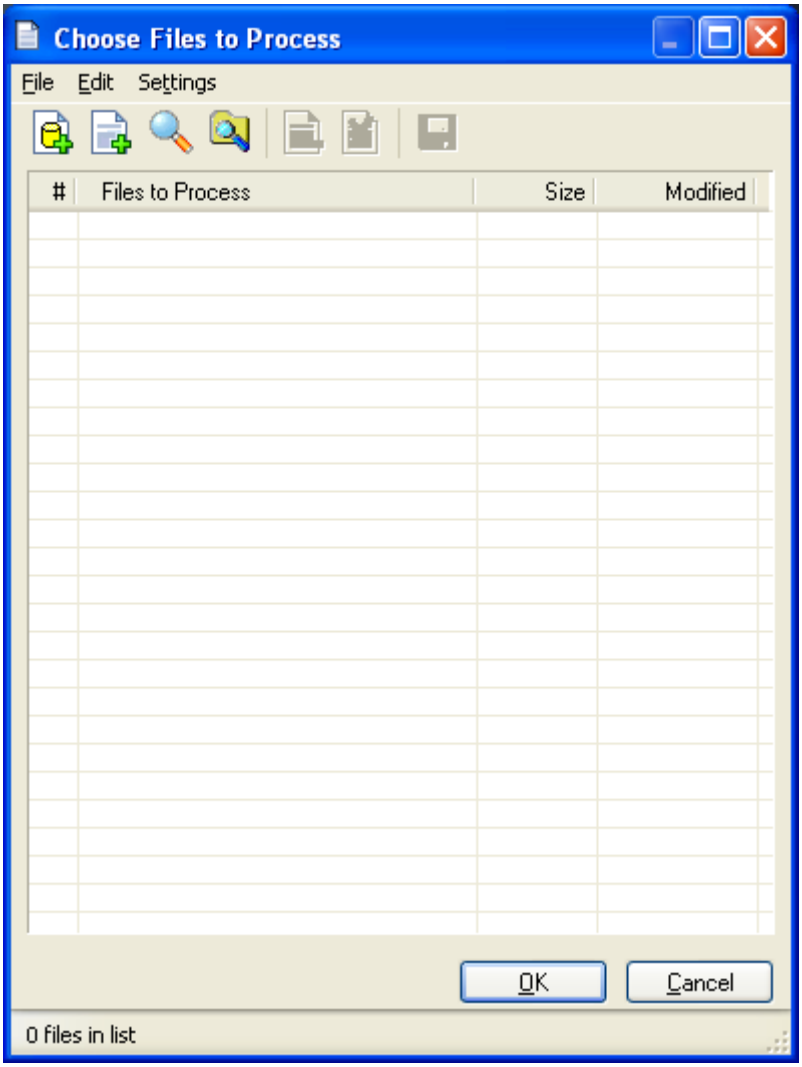
All files must be submitted with a report that explains exactly what operational difficulty you are having with the file(s) in question and everything that needs to be done for us to reproduce the difficulty. We can analyze and remedy only those problems we can reproduce.

If you encounter errors not covered in this manual, send a detailed email to support@AxiomInt.com. Please include as much information about your operating system, the problem that you are experiencing and exactly what you did when this problem occurred. Also make sure to include the version number of MicroStation and *FileFixer* that you are running.

Chapter 17 — Choose Files to Process

Using Axiom's "Choose Files to Process" box

Use Axiom's "Choose Files to Process" box to create a list of files to process. Here is an empty list (no files chosen yet):



The first button shown in this picture — — will only display if you have MicroStation connected to ProjectWise. When this is not the case, this button will not display.

Tip: You can find balloon help for almost any dialog item by holding your mouse over it.

Adding files to the list

There are up to five ways to add files to the list:

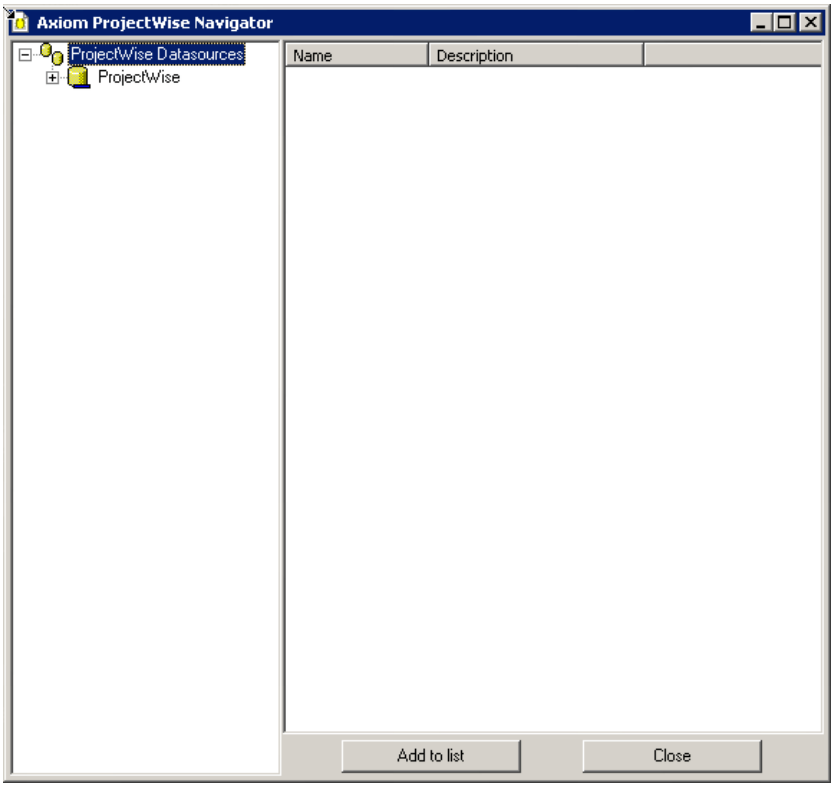
- By browsing for them in a ProjectWise datasource with the “Browse ProjectWise” button . This button is only displayed when MicroStation is connected to ProjectWise.
- By browsing for them with the “Browse for files” button .
- By searching for them with the “Search for files” button , which brings up the “Search for Files” box.
- By dragging and dropping them from Windows Explorer into the “Files to Process” list. You can open a Windows Explorer window using the button.
- By importing an existing ASCII text file with files listed in it using “File | Import File List…”

File and Edit menus

Most of the options in the “File” and “Edit” menus are duplicated along the button bar or on the right-click menu in the “Files to Process” list.

Browse ProjectWise

Click on this button or press {Ctrl-P} to open the “Axiom ProjectWise Navigator” box. You will be prompted to log into a ProjectWise datasource.



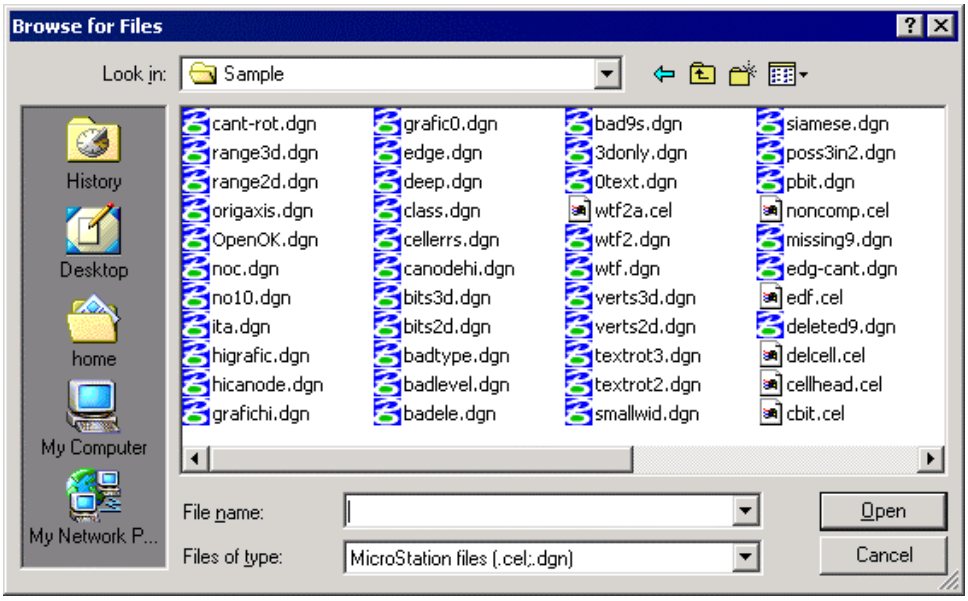
Browse your ProjectWise datasource, select the files you want and press {Add to List}. No files are shown in the above picture but you'll see files once you log into a datasource.

Here you simply select the files you want to process and press {Add to List} to add them to your list of files to process. You can also right-click and choose "Add Files to List" to get the same result.

When you are done adding files to your list of files to process, press {Close}.

Browse for files

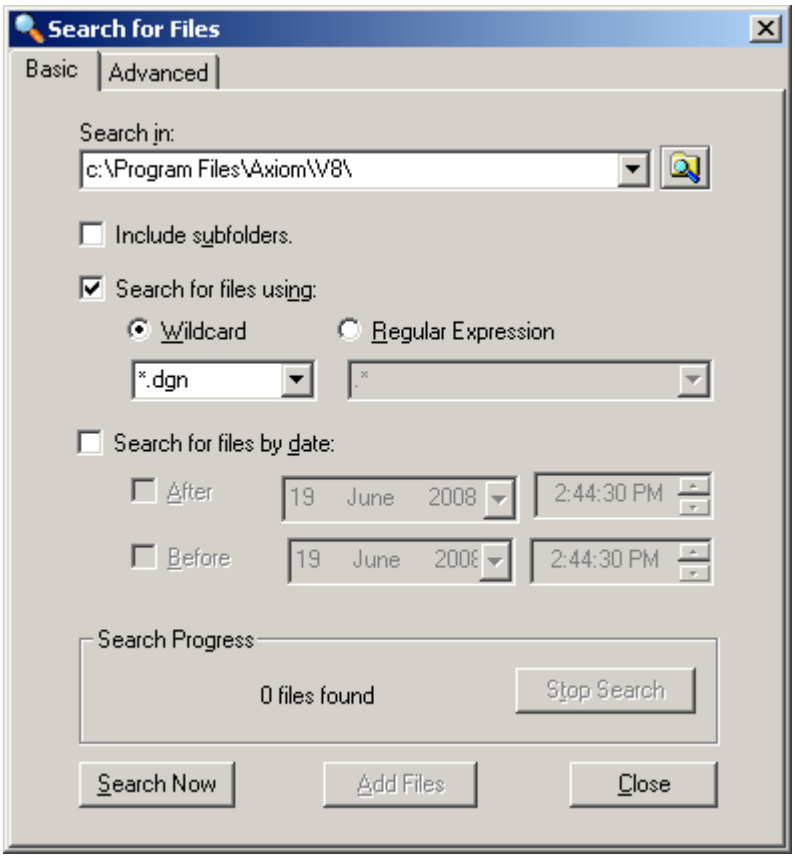
Click on this button or press {Ctrl+O} to open the “Browse for Files” box:



Select one or more files and press {Open} to add them to your list of files to process.

Search for files

Click on this button or press {Ctrl+F} to open the “Search for Files” box (with its “Basic” tab activated):



Search in:

Choose the folder you want to search for files in. Click the “Choose a folder to search” button to browse to the folder you want to search.

Include subfolders.

Enable this if you want to search for files in subfolders too.

Search for files using:

Use this to restrict your search to files with certain names or extensions. Most users search by file extension, such as “*.dgn”.

Search for files by date:

Choose files to process based on their modification date. You can choose to search for files that were last changed before or after a specific date and time.

{Search Now}

Once you have made your file search choices, click {Search Now} to perform the search. When the search is done, you will see a count of the number of files found. Press {Add Files} to add the files to your list of files to process.

{Add Files}

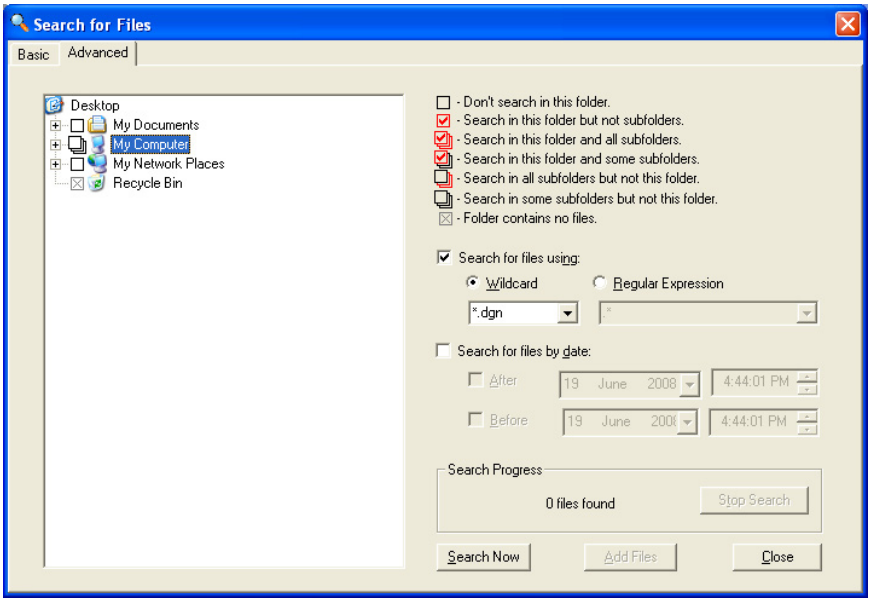
After searching for files, press this button to add the found files to your list of files to process.

{Close}

Press this button to close the “Search for Files” box.

Search for Files: Advanced tab

If you want to search for files in more than one folder at a time, click on the “Advanced” tab of the “Search for Files” box.



Here you can choose any number of folders to search by checking them on in the folder tree.

Drag-and-drop files from Windows Explorer

Click this button or press {Ctrl+E} to open a Windows Explorer window from which you can drag and drop files *or entire folders* into your "Files to Process" list.

Dragging a folder into the list adds the *files* in that folder to the list, not the folders themselves.

Tip: You can drag files into your list from any Windows Explorer window, not just a Windows Explorer window opened by this button.

Dragging folders (not files) into the list

Dragging folders into the list is different than dragging files. Dragging files is direct — the files go right into your list. But since folders could include files you don't want to process, dragging folders initiates a search of those folders. You can then fine-tune the search by adding or removing folders, choosing which types of files to include and limiting the search by file modification dates.

Once you've made your search choices, click {Search Now} to begin the search. When that's done, click {Add Files} to add the found files to the list of files to process.

Tip: When dragging both files and folders into your list simultaneously, the files still get added to your list immediately and a search is initiated for the folders, as usual.

Remove selected files

Press this button to remove the *selected* files from the list.

Remove all files

Press this button to remove *every* file from your list.

Warning: This will clear your *entire list of files.* Use it with care.

Remove Missing Files

This removes any files from the list that cannot be found. This is available only from the Edit menu, there is no dedicated button for it.

ProjectWise Tip: You may be prompted to log into ProjectWise so this operation can determine if any files are missing. If you cancel the ProjectWise login, all ProjectWise files in the list are assumed to be not missing.

Save list to file

Press this button or {Ctrl+S} if you want to save the currently displayed list of files to a text file that you can reuse in the future.

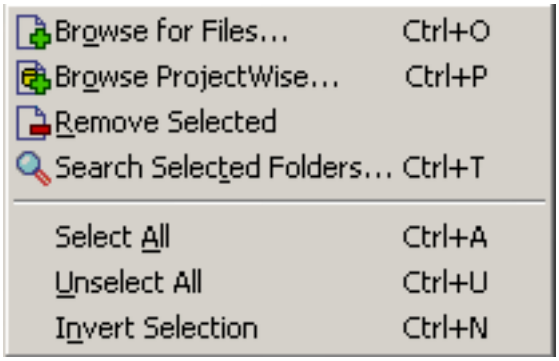
Tip: Import a previously saved file list using the "Import File List..." choice on the File menu or by simply dragging one or more of these .LST files into the list of files to process from Windows Explorer.

Files to Process

The object of this entire box is to put files in the "Files to Process" list.

Right-clicking on the list

Right-click anywhere in the list to see this context-sensitive menu of available actions:



Selecting files in the list

You might want to select one or more files in the list so you can remove them. There are multiple ways to select multiple files in the list.

Using {Shift-left-click}

You can select multiple *adjacent* files in the list by first left-clicking on the first file then holding down the {Shift} key while left-clicking on the last file in the series.

Using {Ctrl-left-click}

You can select multiple files arbitrarily by holding down the {Ctrl} while left-clicking on each one.

Removing files from the list

There are several ways to remove files from the list:

1. Select one or more files and click the {Remove selected files} button.
2. Select one or more files and right-click on one of them, then choose a removal option from the menu which appears.
3. Select one or more files and press the {Delete} key on your keyboard.
4. Click the {Remove all files} button.

Version

Right-click on the title bar and choose “About Axiom File List Processor” to bring up a box containing version information which is useful if you are calling Axiom for support.

{OK}

Once you have chosen all the files you want to process, press {OK} to close the “Choose Files to Process” box.

{Cancel}

If you change your mind, press {Cancel} and any changes you made to your list of files to process will be discarded.

Advanced Commands

The “Choose Files to Process” box is extremely powerful and easy to use. However, for those users who want more raw power, it is possible to create a list of files to process without even picking up your mouse.

Instead of opening the “Choose Files to Process” box, you can type wildcards and other special commands directly in the file specification field in the main dialog box of the host program (for example, *FileFixer’s* “Which Files:” field.)

Using these commands, your list of files to process will be created automatically when you start processing. Or you can open the “Choose Files to Process” box to view and edit your list of files before you begin processing them.

This table describes the most popular of these advanced commands.

Command	Examples	Description
*	c:\work*.dgn c:\work*.* c:\work\project1.*	This wildcard will match zero or more characters anywhere in the file name and extension. It does not work on the path.
?	c:\work\project1.dg? c:\work\project?.d?? c:\work\?project?.dgn	This wildcard will match *exactly one* character anywhere in the file name and extension. It does not work on the path.
/S	c:\work*.dgn /S c:\work\border*.dgn /S c:\work\project?.dgn /S	This command causes subfolders to be searched for matching files. It is usually used with a wildcard.
/L	/L c:\work\list.lst	Follow this command with the name of a list of file to process, including its full path. The list file is a text file with one file per line, including full path on each line, usually with the extension ".LST".

Chapter 18 — Glossary

Specialized terms used in this guide

Note: If you don't find a term here, please check MicroStation V8 on-line help. If that fails, contact us.

Attribute Linkage Any element may optionally contain attribute data or attribute linkage data. This data may consist of a link to an associated database or any other information that pertains to the element.

Attribute data that is not associated with a MicroStation supported database is referred to as a *user data linkage*. A user data linkage may coexist with a database linkage or other user linkages. MicroStation makes no attempt to interpret user linkages; these linkages are maintained when MicroStation modifies an element.

Attribute Offset The V8 version of Index to Attributes (ITA).

In V8 the *attribute offset* defines the offset from the beginning of the element to the first word of the attribute linkage data. If the attribute offset = elementSize, there is no attribute linkage. Abbreviated "attrOffset".

Bit Stands for ***b****inary dig****it*** (0 or 1), the smallest unit of information on a computer. Eight bits make up a byte (or one character).

CAD Acronym for ***C***omputer-***A***ided ***D***esign. A CAD system is a combination of hardware and software that enables engineers and architects to design everything from furniture to airplanes.

CAD systems allow an engineer to view a design from any angle with the push of a button and to zoom in or out for close-ups and long-distance views. In addition, the computer keeps track of design dependencies so that when the engineer changes one value, all other values that depend on it are automatically changed accordingly.

Class Number identifying the class of the element.

0 – Primary

1 – Pattern component

2 – Construction element

3 – Dimensioning element

4 – Primary rule element

5 – Linear patterned element

6 – Construction rule element

Class Map Cell component classes are stored in the cell's class map. In the following examples, bits are counted right to left beginning with 0:

0000 0000 0000 0000 EMPTY (or NULL) — no graphical cell components.*

0000 0000 0000 0001 class 0 — cell contains a primary class element.

0000 0000 0000 0100 class 2 — cell contains a construction class element.

0000 0000 0000 0101 classes 0 and 2 (primary and construction) — cell contains primary and construction class elements.

*A cell's class map can be EMPTY when the cell is composed of nested cells or other non-graphic elements. An element's class is stored in its display header, which only exists in graphical elements (only graphical elements have a class).

Complex Bit When this bit is *cleared* (Clearing a variable means to remove whatever data is currently stored in the variable so that is it set to zero.), this indicates that the element is a simple (non-complex) element. When this bit is *set* (turned on with a value of 1) the element is considered a component of a complex element.

Complex Element A "Complex Element" is a set of elements that are, logically, formed from single component elements (entities), such as lines, arcs and other elements that are 'linked' together to form a single complex element. A "complex element" is a group of two or more elements that are manipulated as a single entity.

Complex Header Element

A "Complex Header Element" is a single element and is the first element of a "Complex Element". This is an element that serves to "introduce" or group a collection of elements. For example, a cell begins with a cell header, followed by the graphical component elements of that cell. Complex headers are not displayable elements.

Compound Document

See "Structured Storage".

Display Header

Each graphical V8 element has a display *header*, which includes its range, properties, symbology and other information only needed by graphical elements.

Element Header

Each V8 element begins with an *element header* containing its type, size, level, ID and other information.

ElementSize

In V8 elements, "elementSize" contains the total number of words in the element.

You must add 2 to V7's words-to-follow (WTF) value to determine the total words in a V7 element — so ElementSize eliminates the necessity to add 2 all the time (easier for programmers).

Element Types

The latest V8 element numbers and types are documented in:

```
...\Program\MicroStation\mdl\include\
mselems.h
```

This file is delivered with MicroStation V8.

Enter-Data-Fields (EDF)

These fields allow a MicroStation user to break up a text string into smaller pieces that are then easily modified. Specifically, an ***E****nter* ***D****ata* ***F****ield* is a one-or-more-character placeholder for future text input. You can insert empty "enter data fields" in a text element and fill in the actual text later. ***E****nter* ***D****ata* ***F****ield's* are particularly useful for variable text such as part numbers or detail numbers.

Floating-Point Number

A number which can contain a fractional part. The term "floating point" is derived from the fact that, in calculating, there is no *fixed* number of digits before or after the decimal point and the decimal point can float. Examples**:** 123.5, 12.35, 1.235.

Graphical Element

A visible element, such as a line or text.

Graphic Group A grouping of elements which are associated by a common *graphic group number*.

Graphic Group Number The Graphic Group Number is used to locate all the elements of a graphic group when performing graphic group manipulations. If this value is zero, the element has no graphic group affiliation. If the value is not zero, the element is included in a graphic group with all other elements having the same graphic group number.

H-bit Stands for the "Hole" bit and is used for closed element types such as shape (6), complex shape (14), ellipse (15), cone (23), b-spline surface header (24) and closed b-spline curve header (27). The Hole-bit indicates whether the element is a solid or a hole depending upon how it is set.

0 = solid, 1 = hole.

The H-bit has different meanings in other element types.

For a cell header (type 2) the Hole-bit indicates:

0 = regular cell

1 = orphan cell (created by "group selection" or application)

For a line (type 3), the Hole-bit indicates:

0 = regular line segment

1 = infinite-length line

For a point string (type 22), the Hole-bit indicates:

0 = continuous

1 = disjointed

Header "Header" is commonly used in several contexts to refer to the beginning or top portion of something.

The first element in a V8 design file is the design file *header*.

Each element has an *element header*.

Graphical elements have a *display header*.

Index-to-attribute linkage

The Index-to-attribute (ITA) linkage is a strictly V7 term which defines the number of words between word 16 in an element and the first word of the attribute linkage data or the next element if no linkage data exists. Word 16 contains the number of words of element data following word 16.

The V7 index to attributes (ITA) is replaced by *attribute offset* in V8.

Mask

A filter that selectively includes or excludes certain values. For example, when defining a database field, it is possible to assign a mask that indicates what sort of value the field should hold. Values that do not conform to the mask cannot be entered.

In MicroStation, it is an area of a reference file that is not displayed.

Nesting and Nesting Level

A single component element that is part of a complex element is considered to be *nested* in that complex element. Complex elements can be nested within other complex elements. Each time a component element is nested within a complex header its nesting *level* would increase by 1.

For example, cell ‘GASKET’ is part of cell ‘FITTING’ which is part of cell ‘FAUCET’. The component elements that make up the cell ‘GASKET’ are also nested in other cells, so therefore its nesting level is 3.

OLE

Stands for ***O****bject* ***L****inking and* ***E****mbedding*, which enables you to create objects with one application and then link or embed them in a different application. Embedded objects retain their original format and links to the application that created them.

For more information on this please search for OLE in MicroStation help system.

Phantom Element

These are elements that are displayed, but cannot be selected or manipulated in any way. This problem comes about when an element that is not part of complex element (such as a cell or a text node) has its complex bit incorrectly set.

Quaternion In 3D design files, the orientation of an element is composed of 9 numbers. This information can be compressed mathematically and represented by only 4 numbers, called *quaternions*. Quaternions are used in some elements to save space. For more, see *Transformation Matrix*.

Range Block The range of each graphical element is stored in its display header. See **Header** entry for more information.

Structured Storage Beginning with MicroStation V8, design files contain multiple internal folders and file, similar to Microsoft Excel or Word documents. This internal organization is called “structured storage”. Documents structured this way are called “compound documents”.

Sub-folders within a compound document are called “storages”.

Sub-files within a compound document are called “streams”.

Text Node This element is a complex header element for groups of text elements.

Text Height & Width Multipliers These multipliers (which are floating point values/numbers) are used to scale a text element against the normal character size to determine the true size of the text element.

Transformation Matrix

A *matrix* is an arrangement of numbers, analogous to a table. For example:

1 0 0
0 1 0
0 0 1

A matrix is used to represent the orientation of an element—are we viewing it from the top, front, right? Let's say the above example represents the *top* view of an element. If we change the element's matrix, the element will be displayed differently. For example, to turn the element upside down and display the *bottom* view of an element, the matrix might be:

-1 0 0
0 1 0
0 0 1

Or, we can change the matrix to increase the size (scale) of an element. The following matrix would double the displayed size of the element in the first example:

2 0 0
0 2 0
0 0 2

A change in size or rotation is called a *transformation*. The matrix used to define such changes is called a *transformation matrix*.

A 3D transformation matrix is composed of 9 numbers, as in the above examples. This information can be compressed mathematically and represented by only 4 numbers, called *quaternions*. Quaternions are used in some elements to save space.

Words

Words are special units of data storage for element information and contain 16 bits or 2 bytes each.

Chapter 19 — We'll Make (Almost) Any Enhancements You Want

We welcome your suggestions

FileFixer is already a powerful program. Nonetheless, we realize that you, a skilled and creative MicroStation user, will think of ways in which it can be improved. We invite you to tell us your ideas.

Axiom didn't become the number one MicroStation third party overnight. For years we've listened to you, the skilled MicroStation professional, and have implemented the features you've requested. Every widely acclaimed product we've ever developed, including *FileFixer*, evolved in this manner.

We'd like to make *FileFixer* perform every function you could ever imagine such a product being able to do. We intend to make you such a satisfied *FileFixer* customer that you can't wait for the next user group meeting so you can tell your friends and associates how much you like *FileFixer* and how strongly you recommend it.

Your suggestions on how to improve *FileFixer* are most welcome.

Chapter 20 — New Features and Fixes

8.14a — 6 August 2014

New! Replace a deleted or badly damaged design file header element.

When a design file's header element is entirely missing or badly damaged MicroStation will not even attempt to open it. Now *FileFixer* can transplant a valid design file header into these fatally corrupted files.

New! Fix shared cells with bad dependency linkages.

Some MicroStation add-on applications use a dependency linkage to find the appropriate definition for a shared cell in a design file. When this linkage is broken the shared cells no longer display. *FileFixer* now finds this problem and can automatically fix it as well.

Improved: Replace a missing shared cell definition from an available cell library.

In the past *FileFixer* would repair a missing shared cell definition with a dummy definition that only displayed the cells name. Now you can have *FileFixer* search available cell libraries for the correct replacement cell definition.

New! Automatically fix elements with too many vertices.

The maximum allowed vertices in a MicroStation V8 element is 5000. When a corrupted element says it has more than this using "fit view" can crash MicroStation. *FileFixer* can now find and corrects this problem.

Additional Improvements

FileFixer now evaluates all non-default models in a cell library when looking for empty models to report.

Fixed a problem that incorrectly reported some reference files as missing in MicroStation XM and earlier versions.

Fixed a problem in Search for Problems mode that incorrectly reported an unused reference file level table (error 267).

Fixed a problem where the model header's dimension (2D or 3D) was incorrectly set when copied or built from a model with a different dimension.

Streamlined report file message output for certain errors.

Fixed a problem that was preventing the Seal-of Approval from being written in "Search for Problems" mode.

8.13a — 12 February 2013

New! *FileFixer* fully tested and certified with MicroStation V8i SELECTseries 3.

A design file can be corrupted with a missing or damaged model header. When this occurs the file will not open. *FileFixer* now fixes this problem automatically.

New! Automatically fix missing or damaged model headers.

A design file can be corrupted with a missing or damaged model header. When this occurs the file will not open. *FileFixer* now fixes this problem automatically.

New! Fix reference file attachments that use slot zero.

FileFixer can now find and repair a reference file attachment that is incorrectly assigned to slot zero. When this happens certain applications, like ProjectWise do not recognize the attachment.

New! Fix damaged Project Explorer data.

FileFixer now finds files that have damaged Project Explorer data. When this problem occurs it can make opening the damaged file take an extraordinarily long time.

New! Remove TriForma modification lock from elements.

FileFixer can now remove the modification lock established by TriForma on certain elements. This lock prevents MicroStation, without TriForma loaded, from modifying these elements. File owners who do not have TriForma are then prevented from changing these elements or copies of them. This new feature gives them a way to fix this problem.

Improved: Find and fix gaps in 2D complex shapes and chains.

FileFixer, in the past, reported gaps in complex chains between lines and linestrings and has reported these as an error 555. This condition is not actually a corruption but rather a condition that some users have wanted to find and fix for various reasons. As such we have reduced the error severity to a suspicious condition and assigned it as error 151. We have also improved the function so that it now applies to both 2D complex shapes as well as 2D complex chains and will find and add lines to fill the gaps between any linear element components they may contain.

Additional improvements

Fixed a license error when using the German language version of *FileFixer*.

Improved structured storage repairs to recover even more data.

Improved corrupt or missing level table handling.

8.12d — 4 May 2012

Changed: Minor update to the licensing system.

Fixed: Fixed an occasional problem when running *FileFixer* in search mode using MSBATCH.BAT.

8.12c — 26 January 2012

Added support for “Raster Hierarchy header” elements (type 92) when testing complex elements for illegal components.

8.12b — 24 June 2011

Enhanced the Seal-of-Approval handling so that any file, without errors, would also get a Seal-of-Approval provided the option to write a Seal-of-Approval is turned on in the “General” settings category.

Enhanced the non-model table replacement function to make sure level, font, text style and line style table indexes for existing elements are correctly updated in the replacement tables.

Fixed a problem that prevented *FileFixer* from running in msbatch mode using MicroStation V8i.

Enhanced *FileFixer’s* handling of line elements when their range was off the design plane but their starting and ending vertices were not.

Made several dialog messages more accurate and easier to understand.

Fixed several formatting problems in the HTM report format.

8.12a — 1 October 2010

Replace a non-model table

New repair settings were added to replace or copy most of the non-model tables contained in design files.

You have the ability to copy or replace the level table, font table, text style table, dimension style table, dictionary table, multi-line style table, line style name table, line style definition table or the filter table.

Copy and replace work slightly different. Replace will only replace the table if it already exists. Copy will replace the table if it already exists or just copy it into the design file if it doesn't exist.

The new error number is 22.

Delete graphical data in the non-graphical data area

New search and repair settings were added to detect and delete graphical data in the non-graphical data area of the V8 design file. These graphical elements can cause MicroStation V8i to lock up or abort. This is now detected (error 819) and handled by this release of *FileFixer*.

Additional enhancements

Error number 702 (Complex bit is on for a non-component element) is no longer incorrectly reported for component elements of cells residing in the non-graphical data section of a design file.

8.11a — 8 January 2010

Corrupt linkage enhancements

FileFixer can now delete siamese linkages, linkages that overlap other elements. The error number was changed from 364 to 764 as this condition can cause MicroStation to abort.

New search and repair options were added to detect and correct missing line feed flags in the linkages of text elements that are part of a text node. This is now detected (error 544) and handled by this release of *FileFixer*.

Reclassified error 635 (“Duplicate text formatting linkage in a text node.”) as a 500-level error (future danger). Its new error number is 545.

8.10a — 4 December 2009

ProjectWise integration

This version of *FileFixer* is now integrated with ProjectWise! *FileFixer* can now check out files from ProjectWise, process them and then check them in automatically before processing the next file.

Additional enhancements

FileFixer now correctly reports missing reference file attachments when the reference file’s file name is prefixed with “file://”.

When duplicate text formatting information is linked to text elements in a text node, the text element can display without “line feeds” (smushed together). This is now detected (error 635) and handled by this release of *FileFixer*.

8.9a — 18 June 2009

Duplicate reference file slot number

Each reference file has a slot number associated with it. Sometimes when slot numbers are duplicated the reference file using the duplicate slot number may not be displayed in MicroStation. This is only a problem when the slot number is duplicated in a model.

New Search and Repair capabilities were implemented in *FileFixer* 8.9a to detect and repair duplicate reference file slot numbers. The reference file associated with the duplicate slot number will have its slot number set to one greater than the highest slot number currently being used in that model.

The error number 314 is reported when a reference file’s slot number is duplicated.

Other enhancements in this release

Internal pointers in the structured storage framework of some customer files were found to point to each other, resulting in an infinite loop (and MicroStation crash). *FileFixer* can now detect and repair this.

Enhanced the search and repair settings for finding and rebuilding model header data to also rebuild model index data. Damaged model index data can make models inaccessible.

Enhanced the reporting of font errors by reporting the font name as well as the font number. Font numbers used in one design file may be different in another design file, whereas font *names* will be consistent.

8.8a — 26 December 2008

FileFixer for V8 version 8.8a is certified for MicroStation V8i.

Due to differences between MicroStation V8i font libraries and earlier font libraries, you might see additional text range errors when V8 files are first processed with *FileFixer for V8*.

By default, MicroStation V8i requires you to take ownership over all changes since the last design history was saved (when Design History is active). When running *FileFixer* for *V8* you may be prompted to take ownership before *FileFixer* continues processing. To eliminate this interruption, set the MicroStation configuration variable MS_DESIGN_HISTORY_OWNERSHIP_WARNING to 0 (zero). You will need to restart MicroStation before this change goes into effect.

MicroStation V8i does not allow *FileFixer* to change level IDs in TriForma elements (to address *FileFixer* error 415).

8.7i — 2 May 2008

Replace the Design File Header Element

The design file header element (type 9) is normally the first element in a MicroStation design file. It stores a variety of vital information.

Design file header corruption can manifest itself in a number of ways:

- Unable to open the design file in MicroStation.
- FIT command behaves oddly.
- After loading the file, MicroStation aborts.
- Text elements do not display correctly.
- Other odd, inexplicable behavior.

FileFixer can replace a corrupt design file header with a healthy one.

The error number is 21.

Note: *FileFixer* will never attempt to replace a design file header unless you explicitly activate the option *Settings* | *Repair Settings* | *Replace the design file header element.* Accordingly, error number 21 will never be reported by *FileFixer*'s Search for Problems command. There is no Search setting which enables or disables error number 21.

Element has Different Dimension than the Model's Dimension

When an element's dimension is different than the model's dimension you cannot always modify that element without first repairing the element's dimension. For example, a 3D line in a 2D model.

New Search and Repair capabilities were implemented in *FileFixer* 8.7i to detect and repair design files which contain elements with a different dimension than the model's dimension.

The new error number 437 is reported when the element's dimension is different than the model's dimension.

Unused Reference File Level Table

Unused reference file level tables just take up space in a V8 design file. They aren't "re-cycled" — each time you attach a reference file a new reference file level table is added to the design file.

New Search and Repair capabilities were implemented in *FileFixer for V8* version 8.7i to detect and repair unused reference file level tables.

The new error number 267 is reported when an unused reference file level table is found.

Duplicate master file level numbers

The level *number* displayed by MicroStation (in the Level Manager dialog) is user-editable and, therefore, not guaranteed to be unique. *Internally* (not viewable by users) MicroStation assigns a *unique* level "code" to each defined level.

By default, *FileFixer* keeps the first duplicate level table entry associated with a master file level *number*. You can optionally keep the last level table entry for a duplicate master file level *number*.

Note: *FileFixer* always retains the "Default" level definition regardless of the repair option selected here. This is a MicroStation requirement.

The search setting that detects this error is *Settings* | *Change Settings* | *Search...* | *Levels* | *Multiple level numbers*.

The error number is 429.

Other Enhancements in this Release

FileFixer reports now include the design file's original file format (when it wasn't V8). This notifies you that the current V8 file was converted from another file format (such as V7 or DWG).

We made a number of refinements to *FileFixer*'s HTML report to improve readability. We also added dozens of additional links from *FileFixer's* reports to the *FileFixer* guide.

Your repaired file can now retain its original name. Of course, if you choose this option, *FileFixer* will require you to select a different output directory (*FileFixer* will not allow your original files to be overwritten).

8.7h — 30 November 2007

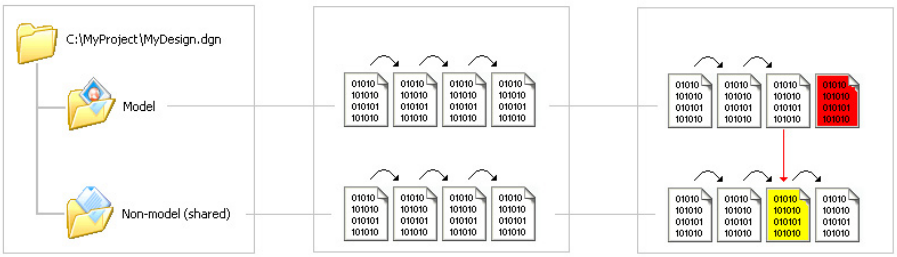
Newly Supported V8-specific Fatal Error

One new category of V8 design file corruption turned out to be so severe it was assigned the new fatal V8-specific error number 898 in this *FileFixer* release. Here's the scoop about error 898:

MicroStation V8 design files are "compound documents" containing subfolders and subfiles. V8 design files contain the same internal structure used by Microsoft to store Word documents and Excel spreadsheets.

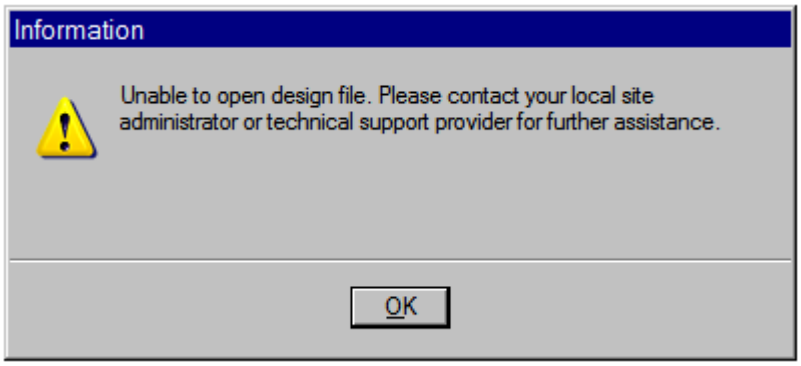
Like all compound documents, MicroStation V8 design files contain an internal "directory" which stores the location of all subfolders and subfiles. If that directory becomes damaged, the relationship between the internal subfolders and subfiles is lost. Without a healthy internal directory, MicroStation will not be able to access the data in a V8 design file — even when the data itself is intact.

FileFixer's newly supported error 898 represents a subtle but treacherous new form of V8 file corruption: two directory entries point to the same data. We coined a new term for this category of corruption: "cross-linked" data [technically, cross-linked streams (subfiles)].



Logical Structure of a V8 File | Physical Structure of a Healthy V8 File | Cross-linked Data

Until cross-linked data is repaired, MicroStation will display the following dialog:

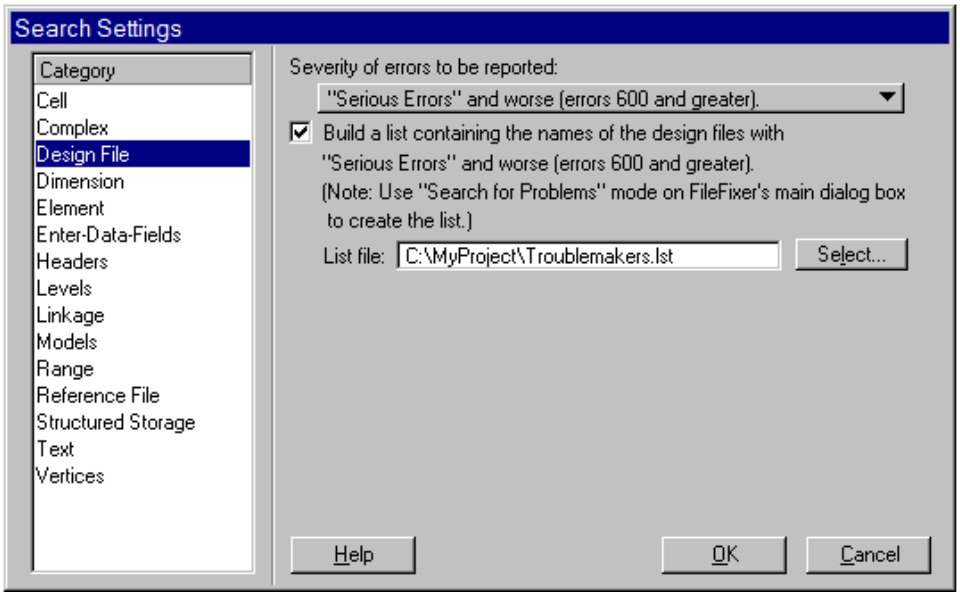

Information
Unable to open design file. Please contact your local site administrator or technical support provider for further assistance.
OK

When *FileFixer* version 8.7h detects error 898 it sorts through the wreckage and repairs the internal directory so all data can again be accessed by MicroStation. If this *FileFixer* repair technology reminds you of Peter Norton's *Unerase* (the popular file recovery utility), you're exactly right.

Other Enhancements in this Release

Torture tests confirm this version of *FileFixer*'s "Search for Problems" command can **Search for Problems** in more than 20,000 design files in a single batch run.

Note: There should never be a need to *repair* thousands of files at a time. Instead, use *FileFixer*'s **Search for Problems** mode and the following settings to generate a list of the *true* troublemakers. Once **Search for Problems** mode has identified the design files with serious errors, use *FileFixer*'s **Automatic Design File Repair** command to repair *just* the files in the list — the true troublemakers. This is the efficient way to ensure the integrity of a huge quantity of project design files.



Until now "Design file header is on the wrong level" was error [311] for both *FileFixer* for V7 and V8. This error is *fatal* for MicroStation V8, so the error severity was changed to number [911] in this release of *FileFixer for V8.*

Zero-length lines (where both endpoints have identical coordinates) are often used to represent a *point* in MicroStation design files. When such points are added as the sole component of a cell, the cell's *name* becomes a convenient label for the point. Cells consisting of a single, zero-length line are very common. *FileFixer* has always reported an error such zero-length lines. This release of *FileFixer* no longer reports error [182] for cells which only contain a zero-length line (point).

Previous versions of *FileFixer* recognized when cells definitions existed in the file but no cell instances existed. However, *FileFixer's* analysis did not include cell instances used as line terminators. This version of *FileFixer* now recognizes when cell definitions are used as cell terminators and no longer reports this as an error.

This version of *FileFixer* has been enhanced to reconstruct a design file when the first non-model stream is corrupted.

8.7g — 31 August 2007

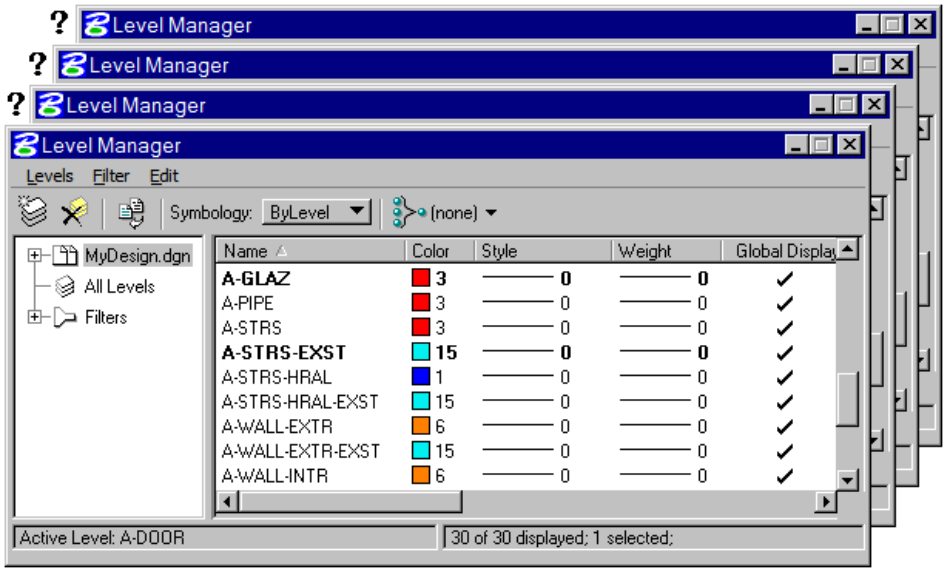
FileFixer 8.7g has been enhanced to detect and repair additional categories of V8 file corruption discovered in V8 design files submitted by MicroStation users.

Four entirely new Search and Repair features had to be invented — these new capabilities are now available in *FileFixer* 8.7g.

Multiple Level Tables (Duplicate Level Definitions)

Level definitions (level names, level symbology, level settings, etc.) are stored in *level tables* in V8 design files. MicroStation displays the contents of a level table in its Level Manager dialog.

There should be *one* master level table for each V8 design file. Surprisingly, analysis of submitted customer files revealed it is not uncommon to find more than one master level table in a V8 file.



Which level table should MicroStation use to display drawing geometry? When you change a level setting in MicroStation's Level Manager, which level table receives the saved changes? Which level table will be active the next time you open the file?

The answers are dependent upon the extent of level table corruption and the version of MicroStation you are using. Prior to V8.5.2.35, MicroStation uses the *first* master level table found in the design file. Starting with V8.5.2.35, MicroStation uses the master file level table with the lowest element ID (usually the original).

Multiple level tables can cause multiple hassles for MicroStation users — have you encountered any of these symptoms?

- Sometimes elements display with the correct level settings and other times the elements display with incorrect level settings.
- Master file level settings appear to match the level definitions for a reference file attachment.
- MicroStation's Element Information tool displays elements with "blank" level names.
- Message Center lists the error message "Detected multiple level tables in file…".

If so, you are fortunate — *FileFixer* can now clean this up automatically for you.

New Search and Repair capabilities were implemented in *FileFixer* 8.7g to detect and repair design files which contain multiple master level tables.

The new error number 630 is reported when multiple master level tables are detected.

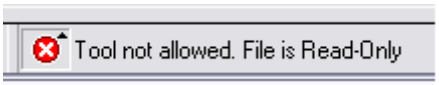
Locked Models

V8 models can be locked. When a V8 model is locked, elements cannot be added, modified or deleted — the model is *read-only*.

Each model in a multi-model file can be locked or unlocked individually.

Models can be locked by MDL applications or by an undocumented key-in command. Models might be locked intentionally.

The misleading message illustrated below appears in MicroStation V8.5's status field when you attempt to modify an element in a locked model. This is the symptom of a locked V8.5 model:



However, the active design file is *not* necessarily read-only — it's the *model* that is read-only!

MicroStation V8.9 is even more confusing. Attempts to add or change geometry in a locked model have no effect (fail) without an error message or explanation.

Locked models can be so perplexing, *FileFixer's* Search and Repair reports now *always* indicate when a model is locked.

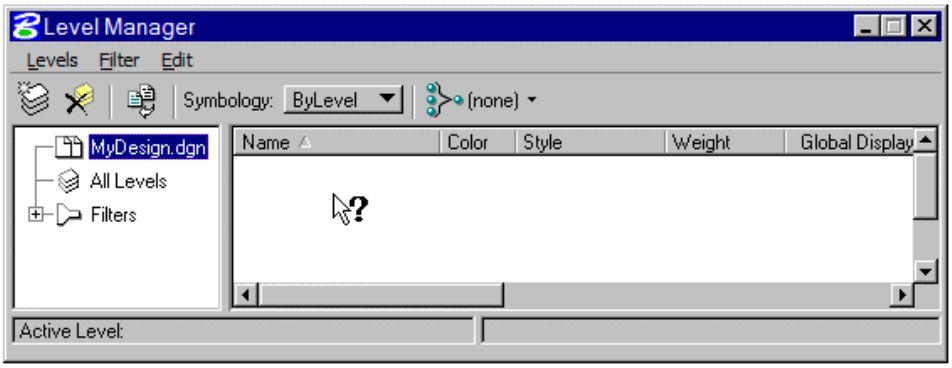
New Search and Repair capabilities were implemented in *FileFixer* 8.7g to detect and unlock locked models.

The new error number 250 is reported when a locked model is detected.

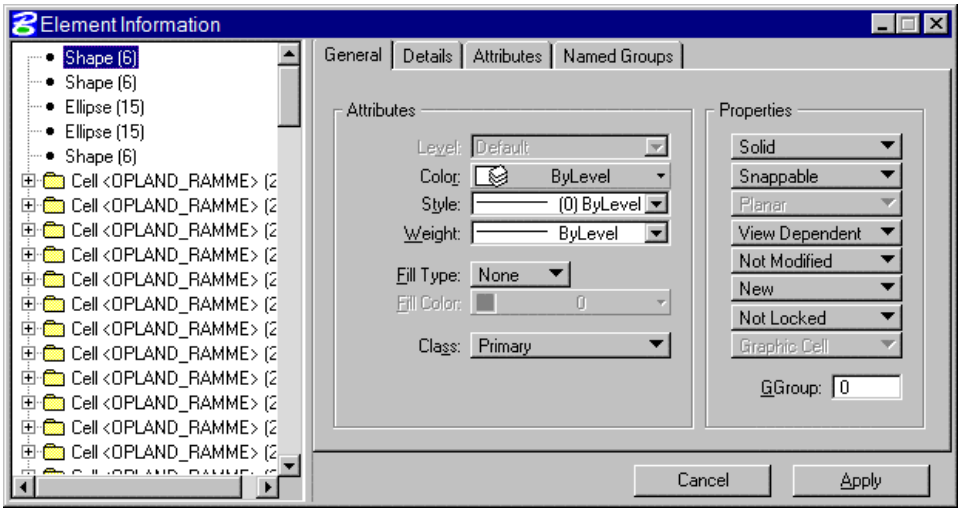
Hidden Levels

MicroStation V8 levels can be *hidden*, which means the level does not appear in any MicroStation level-related tool.

For example, a hidden level is not displayed in Level Manager:



A hidden level is disabled (gray) in MicroStation's Element Information dialog:



Geometry on a hidden level may not display in any view.

Levels can be hidden intentionally by MDL applications, but when this is unintentional (corruption) a hidden level cannot be accessed with MicroStation and the level's geometry might disappear. MicroStation does not offer an "un-hide level" tool.

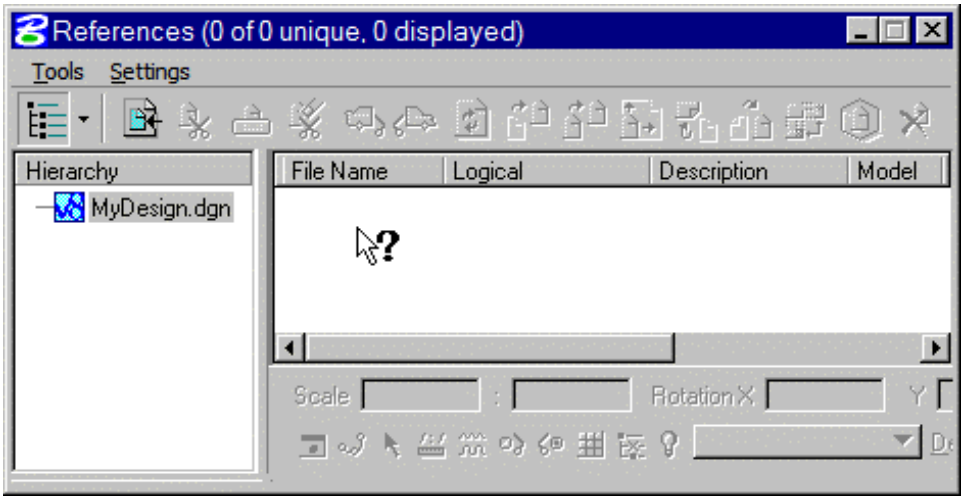
Fortunately, new Search and Repair capabilities were implemented in *FileFixer* 8.7g to detect and display hidden levels.

The new error number 444 is reported when a hidden level is detected.

Hidden Reference File Attachments

MicroStation V8 reference file attachments can be *hidden*, which means the attachment does not appear in MicroStation's "References" dialog and the attachment's geometry is not visible in any view.

Attachments can be hidden intentionally by MDL applications, but when this is unintentional (corruption), the attachment disappears and cannot be retrieved. MicroStation does not offer an "un-hide attachment" tool.



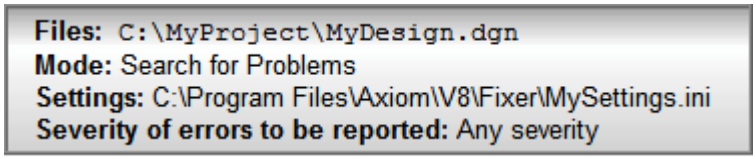
New Search and Repair settings were implemented in *FileFixer* 8.7g to detect and display hidden reference file attachments.

The new error number 443 is reported when a hidden reference is detected.

Additional Enhancements

Error 311 ("Design file header is not on level 8") was upgraded in severity to error 911 because this error is fatal for MicroStation V8.

FileFixer's reports now include the full path to the active *FileFixer* settings file. This documents the settings file which was used to produce the report.


Files: C:\MyProject\MyDesign.dgn
Mode: Search for Problems
Settings: C:\Program Files\Axiom\V8\Fixer\MySettings.ini
Severity of errors to be reported: Any severity

FileFixer Search and Repair processing was optimized to start instantly, regardless of the number of files to be processed.

8.7f — 31 May 2007

All error numbers in *FileFixer*'s HTML report files are now hyperlinks which jump directly to the appropriate paragraph in the *FileFixer* User's Guide. [*Fixer* 8.7e jumped to the correct section or Chapter.]

FileFixer's HTML report now includes convenient hyperlinks which bring all of the following within one mouse click:

- The complete *FileFixer* User's Guide
- Axiom's Support web page
- Axiom's file corruption hot-line (how to submit corrupt files directly to Axiom Support)
- Current versions of all Axiom products
- The latest version of *FileFixer*
- Axiom's web site

FileFixer's QuickStart was updated to introduce newcomers to recent enhancements.

FileFixer's reports are now printer-friendly (automatic when you select *File* | *Print* in your browser).

FileFixer is delivered with templateV8.dgn which is used to salvage V8 files with severe structured storage damage (see description of error 899). Users can now replace templateV8.dgn with a project seed file so that resurrected files inherit project settings perfectly.

FileFixer's new HTML report content was translated to German. To activate German user interface, launch *FileFixer* with:

```
mdl load fixer.ma /l german.rsc
```

Note: full paths are excluded from the above command for brevity. The full command is included in aximenu.cfg.

FileFixer was updated to recognize and handle new element types introduced in MicroStation XM.

FileFixer will not attempt to write its seal-of-approval to a V8 file with error 311 (design file header element is not on level 8) or error 902 (first element of file is not an active type 9 design file header element). XM does not support seal-of-approval updates when these error conditions are present.

Implemented further refinements to *FileFixer*'s detection and repair of severe structured storage corruption (to handle variants of corruption recently found in submitted customer files).

FileFixer's batch mode reports now include only those font-related errors which can be identified without access to a font resource. For example, error 144, "Text element uses non-existent font." is not reported in batch mode.

FileFixer's HTML reports are compliant with current web standards (HTML 4.01 and XHTML 1.0).

MS_LIBRARY_PATH is now initialized to include Axiom base directory and *FileFixer* installation directory. This is a new requirement for version 8.7f.

8.7e — 9 March 2007

FileFixer for V8 version 8.7e includes integrated help and numerous customer-driven enhancements.

Help at the touch of a button

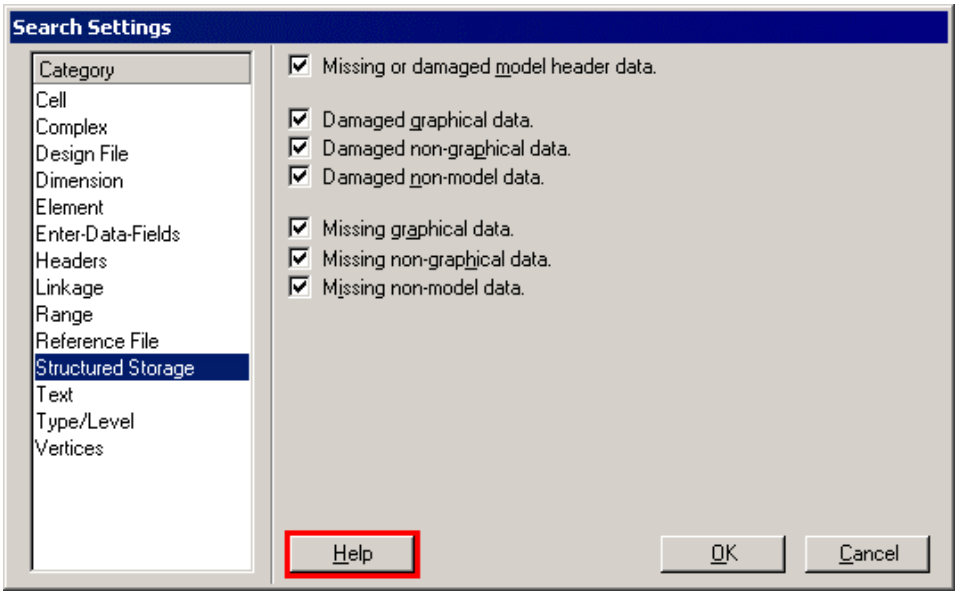
Each HTML report's Report Summary now includes hyperlinks to relevant sections of the *FileFixer* Users' Guide. This allows *FileFixer* users to instantly access help about reported corruptions and what to do to handle.

Report Summary

Quantity	Description	Error Number
4	Element coordinate is too close to edge of design plane or cube.	119
2	Text element doesn't contain any visible characters.	146
76	Z range is not zero in a 2D element.	169
12	Element size is zero in all dimensions.	185
1	Missing graphical data	204
146	Shared cell definition has no shared cell instances.	303
2	Design file header is not on level 8.	311
12	Complex header range is incorrect.	320
70	Range of arc or ellipse is incorrect.	324
2	Class of element is illegal.	333
32	Reference file is missing.	343
6	Element coordinate is beyond edge of design plane or cube.	619
10	This type cannot be a component of that type.	721
1	Range of arc or ellipse is incorrect.	724
6	Range of text element is incorrect.	725
1	Missing header data	801
1	Internal storage in the file is corrupted.	899

Errors may be reported for an element more than once (in different repair steps).

FileFixer offers a *lot* of options to precisely control how to search for and repair corruptions in your project files. When the going gets tough and you need to repair a troublesome file *fast*, you need immediate access to help. New with this version of *FileFixer*, each settings dialog now includes a <Help> button which immediately jumps to the relevant section of the *FileFixer* User's Guide — instant access to the help you need.



Customer-driven enhancements

Since its inception, *FileFixer* has evolved in response to customer requests and from analysis of repeating file corruption trends. This continues with 8.7e — this version includes a number of customer-driven refinements, such as:

New Search and Repair settings were implemented to detect and delete a shared cell instance when nested within a shared cell definition with the same name. This corruption can prevent MicroStation from opening a design file. The new error number is 733.

New to V8, certain elements can “safely” and *legally* reside outside the design plane or cube without impacting FIT or other MicroStation commands. *FileFixer* now recognizes these exceptions — truly troublesome range errors are still identified and reported.

FileFixer Search and Repair processing now starts much faster when <Start> button is pressed.

FileFixer’s bottom progress bar now reflects the *number* of files being processed rather than a count of internal sub-files.

FileFixer now has a new “Do not check for invisible elements” option (ON by default) on the MicroStation Application settings window for TriForma (Bentley’s architecture add-on to MicroStation). Invisible elements are often found to be unwanted in “plain” MicroStation files submitted by customers, so their existence is reported by default. But invisible elements are common in TriForma files (necessitating this new TriForma-specific option).

German Search and Repair categories are now sorted in the same sequence as the English category lists (direct correlation with English documentation).

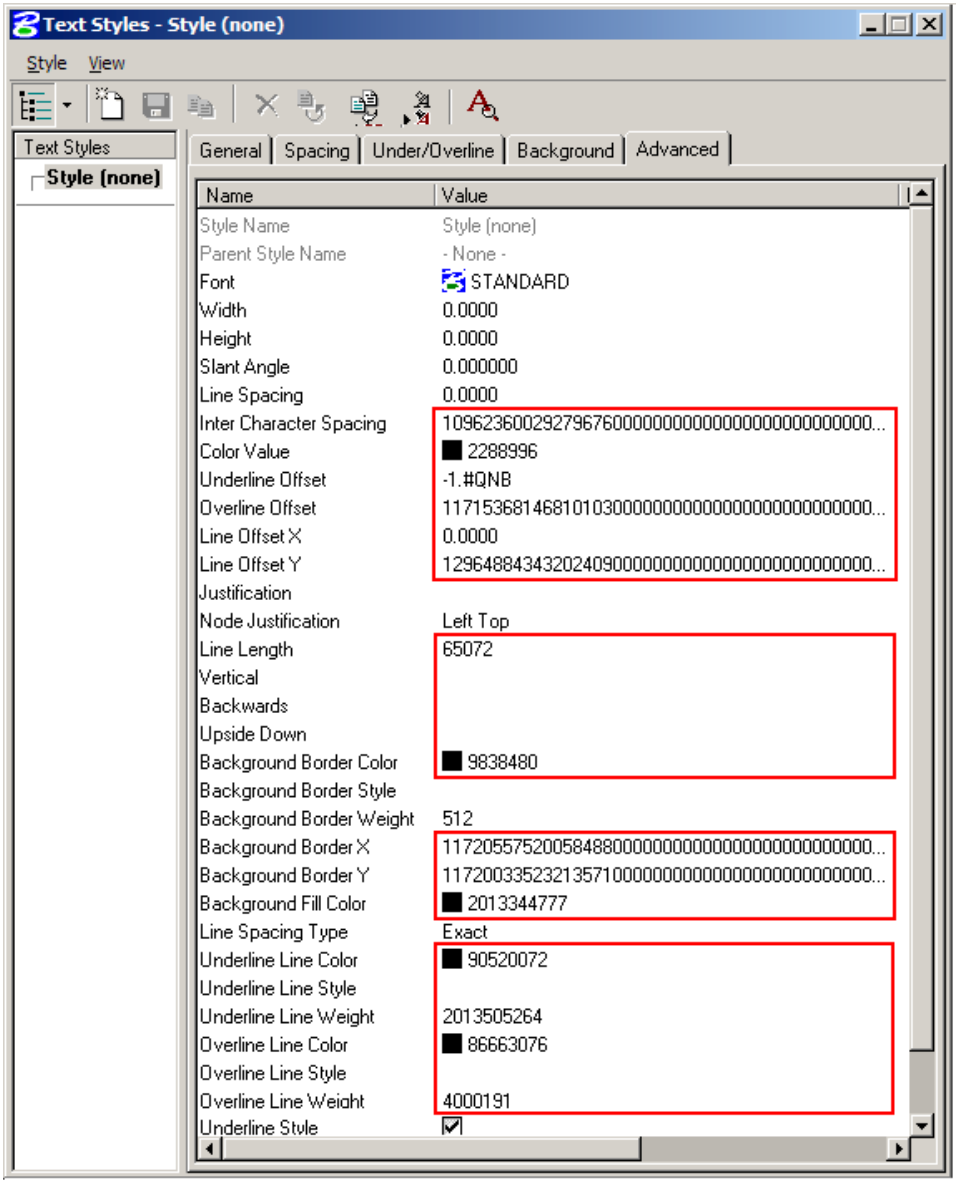
Text Style corruption

The default text style is saved in every design file's TCB and is used when no other text style is defined and active. When the default text style settings become corrupted, merely attempting to place text (with default text style) may cause MicroStation to abort.

The new error number reported is 745.

When *FileFixer* detects and reports this form of corruption, MicroStation users can enter legal values by selecting *Element* | *Text Styles* from MicroStation's main menu, then the "Advanced" tab.

The following screen image illustrates *multiple* instances of corrupted values for the default text style (none) — each highlighted region contains one or more corrupted values:



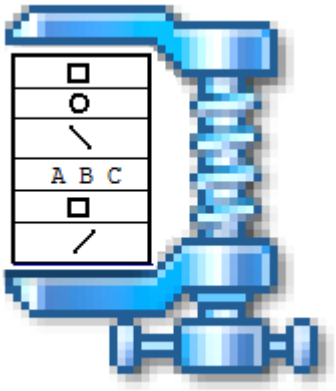
Advanced Technical Data: A default text style is saved in every V8 design file's TCB (terminal control block). Custom text styles, if present, are stored separately in a text style table in the non-model portion of the file.

8.7d — 24 November 2006

Once again *FileFixer for V8* has been enhanced to detect and repair new forms of V8 file corruption frequently appearing in files submitted to us by MicroStation users. These new corruption categories are described here for the first time.

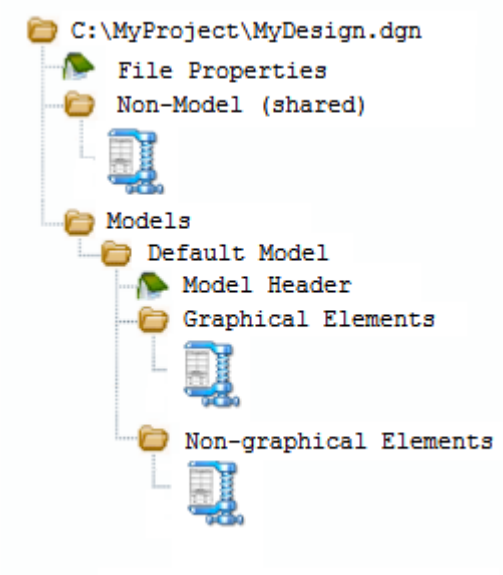
V8 elements are compressed.

Unlike V7 elements, V8 elements are compressed (much like **`.zip`** archives) to reduce file size.



V8 design file organization is new.

MicroStation V8's compressed elements are stored in subfiles within subfolders.



Structured Storage directory corruption

Each V8 file contains a *directory* of the structured storage subfolders and subfiles within it. Analysis of MicroStation V8 user files has revealed that the data within a structured storage file can be healthy, but *inaccessible* because the directory itself is corrupted. This version of *FileFixer for V8* has been enhanced to detect and repair damaged directory entries.

New Search and Repair breakthroughs

When the structured storage framework of a file is damaged, there is *no way* to access the elements within.

Similarly, when the compressed portions of a V8 file are damaged, there is no way to access the elements within.

V8's structured storage file organization and element compression normally work fine, but when one or the other or *both* become corrupted, there is no way to access the elements nestled within. Analysis of customer files reveals these *are* the most common categories of file corruption unique to V8.

This version of *FileFixer for V8* includes completely new analysis and repair features to handle these V8-specific categories of corruption.

In fact, *FileFixer for V8* is the only application available *anywhere* that can automatically repair these new categories of V8 corruption.

Plus timing tests confirm this version of *FileFixer for V8* repairs files up to 2X faster than the previous release.

FileFixer for V8 version 8.7d was tested with MicroStation V8 8.5.2.35 and XM 8.9.2.77.

8.7c — 28 August 2006

FileFixer for V8 versions up to and including 8.7b provide solutions to the most commonly occurring, disastrous new forms of V8 file corruption.

Version 8.7c gets *personal* — no previous release of *FileFixer for V8* addressed more client requests.

Some of the customer requests addressed in 8.7c are pretty geeky. Enhancements suitable for mass consumption follow.

- By popular demand, *FileFixer-Lite for V8* is now available for analysis of V8 files.

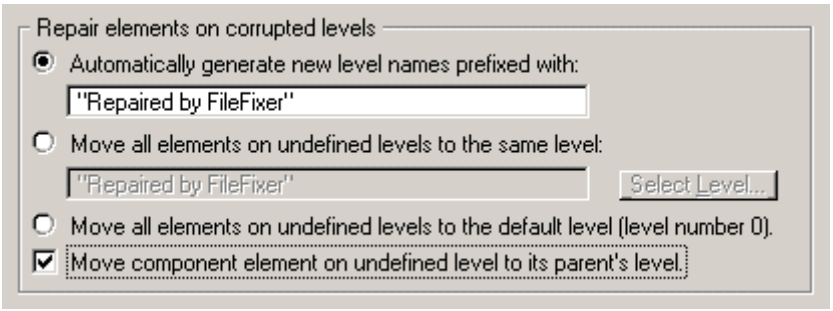
Now CAD managers, designers, drafters — any MicroStation V8 user — can find out, with the click of a button, if MicroStation V8 files have fallen prey to file corruption. *FileFixer-Lite* reports a summary of all forms of file corruption found. A full *FileFixer for V8* license is required to repair corrupted V8 design files.

FileFixer-Lite for V8 report excerpt:

Report Summary

71	--	Suspicious – This element may not violate any “official” MicroStation rules, but something about it doesn't look quite right.
154	--	Warning -- This element violates a rule of MicroStation file construction, it may eventually cause problems.
12	--	Serious Errors -- It is likely the element will manifest problems of some kind. It is important to fix such errors to avoid problems.
2	--	Fatal error -- It is unlikely that any MicroStation related processor will process this file in its entirety without a severe error.

- *FileFixer for V8* now displays its report in HTML format by default. The ASCII report option remains available.
- More sample files are now delivered with *FileFixer for V8* to illustrate the various forms of V8 corruption.
- 8.7c offers the option to move component elements to their parent’s level.


Repair elements on corrupted levels
Automatically generate new level names prefixed with:
"Repaired by FileFixer"
Move all elements on undefined levels to the same level:
"Repaired by FileFixer"
Select Level...
Move all elements on undefined levels to the default level (level number 0).
Move component element on undefined level to its parent's level.

- A linear element with an end point off the design plane now retains its original angle when repaired.
- *FileFixer for V8* can now repair all common graphical element types when they overlap or reside entirely beyond the design plane edge.
- 8.7c now reports both the error number and error message for seals-of-approval created by the V7 version of *FileFixer*.
- Clarified a number of error messages in *FileFixer’s* report to answer common questions about file corruption.

- Updated the Error Messages chapter in the user's guide to include the exact menu location of each Repair setting and each setting's default value. *FileFixer for V8* has a *lot* of options — it's easier than ever to find the exact option needed to salvage your design file.
- The size of each design file before and after repair is now included in each report.
- Added support for the new configuration variable "FIXER_DIRFORBKPFILES". Use this to define the location where *FileFixer* creates backup files. This configuration variable overrides the *General | Directory for backup files* setting.
- Improved range repair algorithm for cones and curves. Now handles some unusual cases found in customer files.

FileFixer for V8 version 8.7c was tested with MicroStation V8 8.5.2.35 and XM 8.9.2.52.

8.7b — 8 June 2006

Added the option to "Move all elements on undefined levels to the default level (level number 0)" when repairing elements on undefined levels.

Note: Previous versions of *FileFixer for V8* offered the level repair option "Do not repair." However, when such corrupted elements are modified, XM reassigns them to the default level (whether you want this or not). *FileFixer for V8* now automatically generates new level names for such corrupted elements by default. The benefits of this approach are described below under the release notes for version 8.6b.

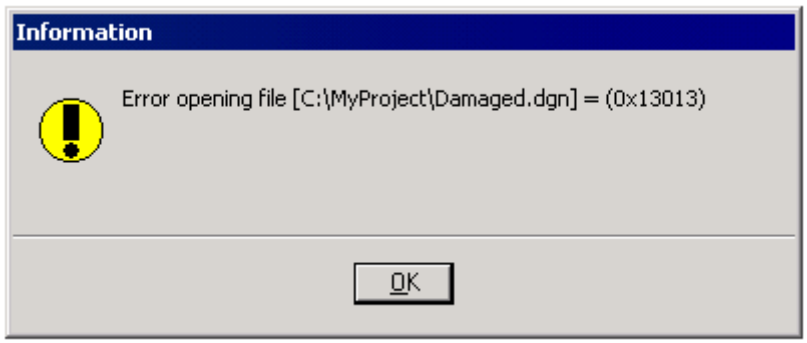
A level table must exist before repaired levels can be restored in a file. *FileFixer for V8* now has the ability to create a missing level table (to which repaired levels can be added).

Certified for MicroStation V8 XM edition version 08.09.02.52.

8.7a — 31 March 2006

When the internal framework of a V8 file is corrupted, the design geometry in the file can be completely inaccessible. Internal structure damage is the most frequently occurring new form of V8 file corruption.

The symptom? The V8 file cannot be opened and a *helpful* error message displays:



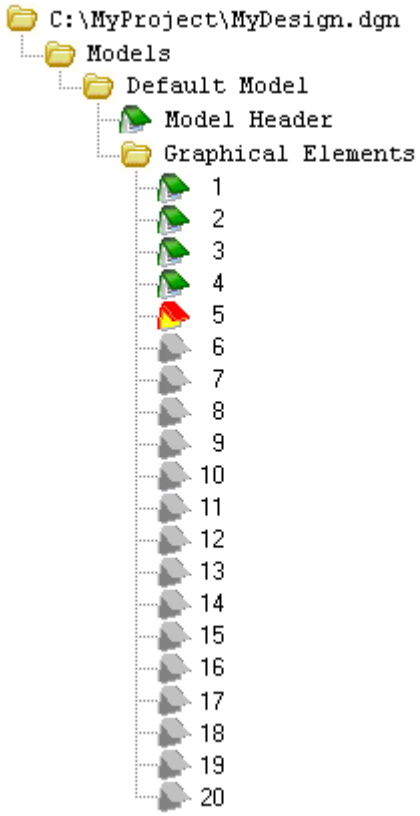
FileFixer has been enhanced to handle this new category of V8 file corruption.

Completely new file recovery techniques had to be developed. The new approach is analogous to the famous Norton *Undelete*, which salvages “deleted” files from a hard drive — even when the file is stored in fragmented pieces.

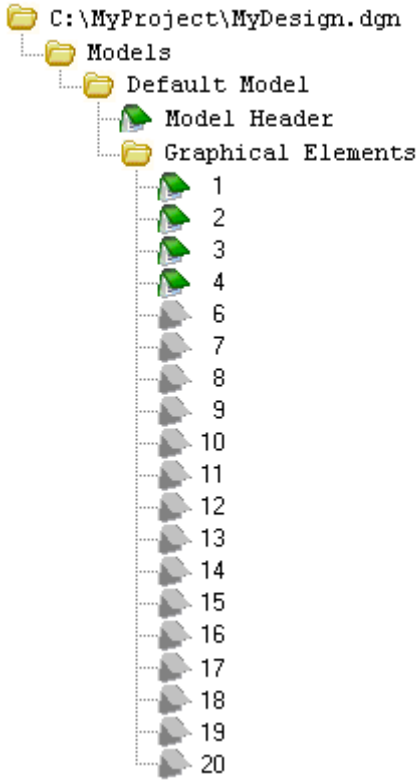
When *FileFixer* detects severely damaged internal file structure, it reads through the entire file looking for healthy pieces of the original file. The healthy pieces are recombined into a healthy V8 design file.

But internal structure damage can wipe out large chunks of a design file without warning — no obvious error dialog. One piece of bad data can prevent the *remainder* of your model from displaying.

In the following illustration, subfolder #5 is bad and elements stored in the remaining subfolders 6-20 (containing more than 75% of the drawing's geometry) are not displayed:



And, even more insidious, a *missing* subfolder can also prevent remaining geometry from displaying. In the following illustration, subfolder #5 is *missing*. Elements in the remaining subfolders 6-20 will not display:



In a V8 file, a *model header* is analogous to a cell header in a V7 cell library. The model header stores the model's name, description, etc.

The previous version of *FileFixer for V8* could rebuild a *missing* model header, but it could not rebuild a *damaged* model header. *FileFixer* 8.7a can now reconstruct a damaged model header.

FileFixer is now able to calculate and report the number of lost elements in a damaged portion of a V8 file. Previously, there was no way to know how much of the drawing was at risk! The true extent of V8 file corruption is now disclosed for the first time.

This version of *FileFixer* was certified for MicroStation V8 versions 8.1.2.15 and 8.5.2.35.

8.6b — 6 December 2005

When a V8 file's level table is damaged or missing, graphical elements become "stranded" on levels that do not exist.

Previous versions of *FileFixer for V8* moved all such "stranded" elements to a new user-defined healthy level (or any level that still survives in the file's level table). This approach recovered all such corrupted elements, but the corrupted elements were moved to the same level.

This version of *FileFixer* offers a significantly improved remedy for damaged or missing level tables. Even when a level table is corrupt or entirely missing, level information remains in each element — *FileFixer* uses this element information to repair the level table.

When a damaged or missing level table is detected, *FileFixer* examines all "stranded" elements and moves each stranded element to a new level, but now retains the original level relationships for all elements. Elements that previously occupied a particular level are reunited with all other elements which also occupied that level. New level *names* are auto-generated and can be used as is or can be easily edited in MicroStation's Level Manager dialog to restore their original labels.

In the V7 version of *FileFixer*, level corruption is identified by error number 115. The impact of damaged or missing level tables in V8 is severe, so error number 115 has been retired in *FileFixer for V8* and replaced by error number 415 to properly reflect the severity of this corruption in V8. Since level tables do not exist in MicroStation V7, error 115 remains unchanged in *FileFixer* for V7.

In addition to the level table enhancements described above, this version of *FileFixer* can now successfully traverse completely unreadable non-model data found in severely damaged files submitted by Axiom clients.

8.6a — 18 August 2005

The tool used to create a list of design files and cell libraries to process has undergone a dramatic improvement — it has been completely rewritten. Its user interface is easier to use than ever and it now supports drag and drop of files from Windows Explorer. It is also faster at searching for files to process.

Added the new Search option *Z range is not zero in a 2D element.* The error reported is "169" and the details of this error are described below:

169 — Non-zero Z range found in a 2D element.

The existing Repair option *Repair range of displayable elements* has been expanded to correct this problem. These Search and Repair options are on by default.

FileFixer now reports when it finds more than one copy of its required DLL files in certain places on the system and instructs the user to delete duplicates.

FileFixer now checks the version number of required DLLs to ensure the correct version (or later) is installed.

8.5b — 24 February 2005

Enhanced *FileFixer's* "Set modification date and time of the fixed file to that of original file" option to be used when working on the original file and creating a backup file. In this case, the backup file will always retain the date and time of the original file, and the

modified original file's date and time will be optionally maintained or set to the current date and time.

Enhanced *FileFixer* to be able to continue checking and repairing V8 design file corruptions after a damaged data area has been found. Previously, *FileFixer* would quit processing the V8 design file after a damaged data area was found. *FileFixer* can now delete these damaged data areas.

FileFixer can now detect and correct when the design file header element was not on level number 8 (error 311). When this is the case MicroStation is unable to open the design file.

MicroStation V8 has a feature that allows elements to be set "invisible". These invisible elements are not displayed in MicroStation's GUI or plotted. If you also use Cimmetry's AutoVue application then these elements *are* displayed, which makes it confusing when you compare the design file's display between MicroStation and AutoVue. We have added a new *FileFixer* error number (240). *FileFixer* can correct these invisible elements by either making them visible or deleting them. These settings are turned OFF by default.

The "Report File" dialog box was modified to include all the report file settings. Previously, they were on a couple of different dialog boxes.

The "Options" menu was renamed "Settings".

The organization of the "File" and "Settings" menus was greatly improved.

The "General" and "Report File" Settings categories were carefully reviewed and simplified to make the most common *FileFixer* options more easily accessible.

8.5a — 10 September 2004

Enhanced *FileFixer's* range repair handling of error 619 to move an element which has at least one coordinate beyond the edge of the design plane back onto the design plane. The element coordinate is moved within approximately 5% of the edge of the design plane. This allows the user to either move the element to its correct location or delete it. After repairing error 619, users can now snap to the element, move it or delete it.

FileFixer's processing is so CPU intensive the operating system is not always able to keep the screen refreshed. This is particularly true with Windows XP. *FileFixer* now gives the operating system a slice of time, once per design file being processed, to refresh the screen and perform other system tasks.

Various internal optimizations have greatly increased *FileFixer* processing speed. In our test cases, repair processing time was reduced by 5X or more.

Shared cell nesting errors are reported as error 721 -- "An element of this type cannot be a component of a complex header of the specified type". *FileFixer's* report now includes the parent and nested cell's names (in addition to the element file positions and IDs) for each 721 error reported. It is often easiest to identify and work with cells by name.

8.4b — 7 May 2004

FileFixer now creates temporary files in the directory defined by the Windows environment variable TEMP. See the section “Alternate temporary files directory” for details.

Nested shared cell definitions are now squeezed out of shared cell definitions (error 303).

8.4a — 10 April 2004

Added new error number 899 to detect and report when the internal structure of a V8 file is so severely damaged the file cannot be opened as a structured storage file — this condition prevents access to any data in the file.

Enhanced *FileFixer’s* handling when an element contains a level that is not present in the level table (error 115). Analysis of many client files with damaged level tables allowed us to implement a more solid, comprehensive repair algorithm.

When processing large numbers of files, we encountered cases where the total for one or more *FileFixer* errors detected exceeded 65,535 (the maximum value representable by the existing 16-bit counters). To accommodate these unprecedented error totals, *FileFixer* counters were increased to 32-bits. Now more than 4 billion errors can be tallied for *each* error number.

Internal optimizations were done to increase *FileFixer’s* processing speed. These optimizations were implemented to compensate for the additional processing time required by the accumulation of recent *FileFixer* enhancements.

FileFixer now handles V8 text elements up to V8’s limit of 1028 characters.

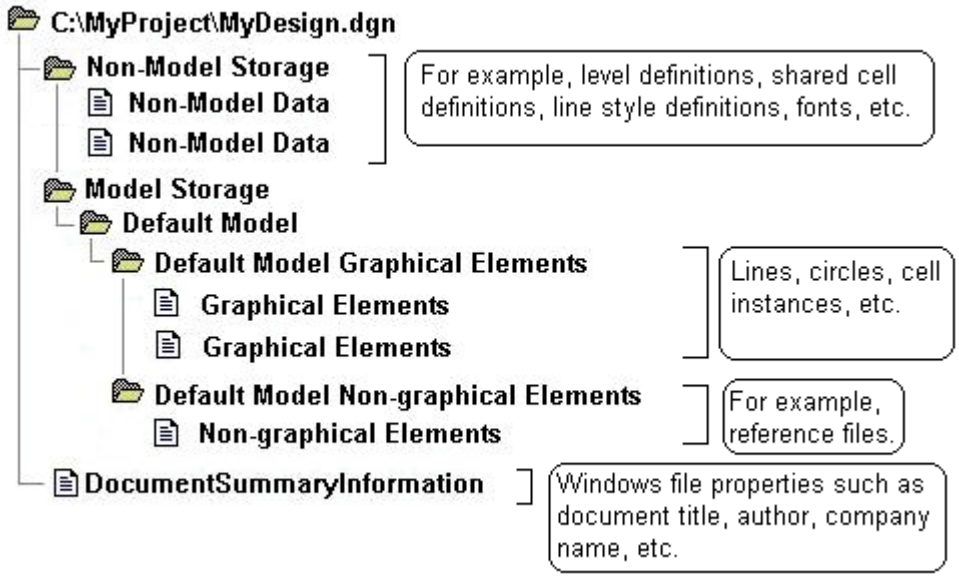
FileFixer displays a history of recently selected files under its “File” menu. File names which exceed 80 characters are now abbreviated in the list.

You can display *FileFixer’s* ASCII report using any editor (Notepad.exe by default). Your selected editor is now a savable setting. This allows you to use WordPad.exe or any other editor of your choice to display *FileFixer’s* ASCII reports.

Changed error number 162 to 62 (“This is the first element indicating this file contains TriForma linkage.). Informational message 62 is not an error and is now excluded from the Report Summary.

8.3a — 30 January 2004

Each MicroStation V8 file consists of multiple internal directories and sub-files. This data organization scheme, also used by Microsoft Word and Excel, is called *structured storage*. When saved to disk, MicroStation V8 design file elements are compressed and nested in the sub-files.



Structured storage in a V8 design file (simplified for clarity).

The structured storage system is normally reliable and robust, but as with any computer application and file type, data can become corrupted.

If one of the structured storage folders or sub-files is damaged, the data therein might become inaccessible, even when other portions of the compound file are intact.

Analysis of MicroStation V8 files submitted to Axiom revealed that structured storage errors are one of the most serious V8-specific problems occurring in production. This is a new phenomenon with V8 files. If you're working with V8, you might have encountered structured storage errors without realizing it.

Typical symptoms of structured storage errors are:

Missing elements.

"MicroStation Problem Notification" dialog boxes.

Models which cannot be selected from MicroStation's model's dialog.

In response to customer demand, *FileFixer for V8* was enhanced to detect and handle the most serious structured storage problems.

Search and Repair enhancements

Error	Search Setting	Repair Setting
801	*Structured Storage* | ***Missing*** *or damaged model header data.*	*Structured Storage* | *Rebuild damaged model header data.*
	Structured Storage | ***Missing*** *or damaged control data.*	None.
	Structured Storage | ***Missing*** *or damaged graphical data.*	None
802	*Structured Storage* | *Missing or* ***damaged*** *model header data.*	*Structured Storage* | *Rebuild damaged model header data.*
	Structured Storage | *Missing or* ***damaged*** *control data.*	None.
	Structured Storage | *Missing or* ***damaged*** *graphical data.*	None.

For much more information about these new features, see:

Error Messages: Fatal V8 specific Errors.

Repair Settings: Structured Storage.

Search Settings: Structured Storage.

This version of *FileFixer* was certified for MicroStation V8 version 8.1.1.9.

FileFixer versions 8.2a and 8.3a were certified simultaneously.

8.2a — 20 December 2003

Our objective was to provide additional Search and Repair solutions for problems found in client V8 files. *FileFixer* enhancements include:

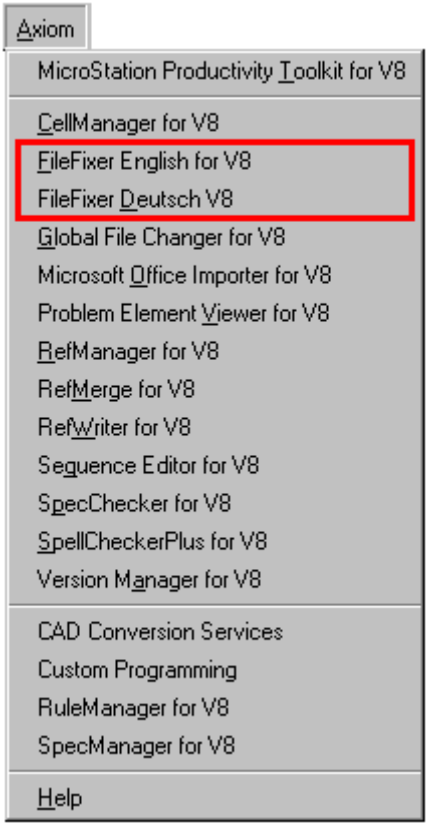
Search and Repair enhancements

New features are indicated in **bold**.

Error	Search Setting	Repair Setting
115	**Bad element level (level is not present in the level table).**	Move elements on unacceptable levels to level: _____.
119	**Element is too close to edge of design plane.**	Delete elements with the following error numbers.
191	**Text element has more EDFs than characters**	**Correct EDF/text length mismatches.**
302	**Type 1 (cell library header) elements are not allowed in design files.**	Remove illegal element types.
333	**Element class is illegal.**	**Set illegal class to ___.**
343	Missing reference file.	**Delete missing reference file attachment.**
364	**Siamese attribute linkage (linkage overlaying the next element).**	**Fix element size warnings.**
380	**Illegal element type detected.**	**Remove illegal element types.**
514	**Attribute offset error.**	None.
560	**Text string overflows the element's length.**	**Correct the number of characters field in text element.**

Additional enhancements

You can now load *FileFixer's* foreign language interface from the Axiom menu. A German interface is available now. Please let us know what other language(s) you need:



Now reports missing *raster* reference files (error 343) and includes setting to delete missing raster references.

We increased *FileFixer's* capacity for maximum file name length from 128 to 256 characters.

FileFixer now reads the level table during prescan to determine which levels are valid for the elements in the file. This supports the Search setting "Bad element level" and Repair setting "Move elements on unacceptable levels to level: ____" (error 115).

FileFixer was enhanced to properly interpret linkages which have "number of words" or "number of bytes" (new to V8) linkage size fields.

Customers presented us with files containing element coordinates that were *off* the design plane. Further analysis revealed that MicroStation only uses a tiny portion of the potential range of floating point numbers (to preserve precision), but some V8 elements contained floating point values outside the range recognized and supported by MicroStation. Accordingly, *FileFixer's* handling for error 119 was modified to recognize element coordinate values which are beyond the edge of the design plane (or cube). [This phenomena is new to V8 — V7 element coordinates, represented by integers, could not exceed the design plane edge.]

FileFixer recognizes new element types introduced in MicroStation V8.5.

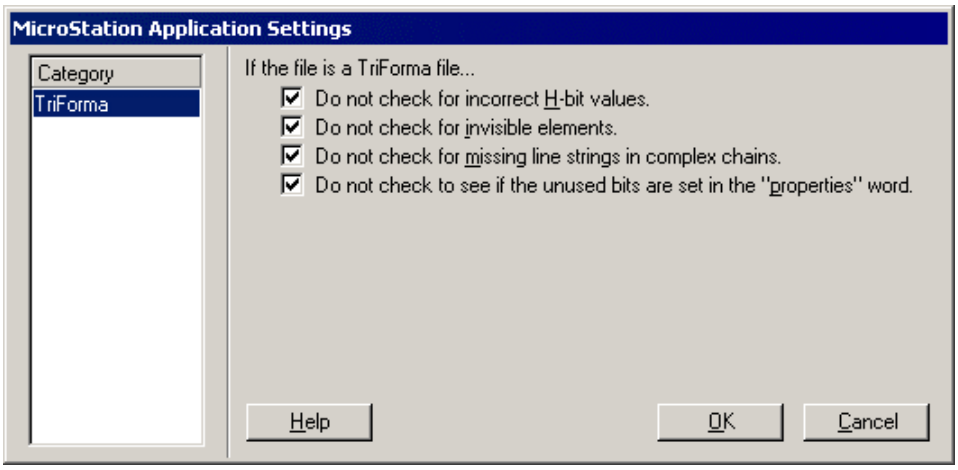
This version of *FileFixer* was certified for MicroStation V8 version 8.1.1.9.

FileFixer versions 8.2a and 8.3a were certified simultaneously.

8.1a — 25 July 2003

FileFixer 8.1a enhancements include:

More TriForma-specific defaults


MicroStation Application Settings
Category
TriForma
If the file is a TriForma file...
Do not check for incorrect H-bit values.
Do not check for invisible elements.
Do not check for missing line strings in complex chains.
Do not check to see if the unused bits are set in the "properties" word.
Help
OK
Cancel

"MicroStation Application Settings", "TriForma" settings dialog box.

Do not check for a cell nested within itself when the cell name is blank

FileFixer normally reports error 383 when it detects a cell nested within itself (a self-referencing cell). TriForma files often contain a variation of this: unnamed cells nested within unnamed cells — a legitimate construct in TriForma files.

When a TriForma file is detected, *FileFixer* will not report *or repair* error 383 unless you explicitly disable this setting. This ensures *FileFixer's* report emphasizes the most severe errors.

Do not check for incorrect H-bit values

FileFixer normally reports error 304 when an element's H-bit is set to "1" when it should not be. This error is common, but not serious.

When a TriForma file is detected, *FileFixer* will not report or *repair* error 304 unless you explicitly disable this setting. This ensures *FileFixer's* report emphasizes the most severe errors.

Do not check for missing line strings in complex chains

FileFixer normally reports error 555 when it detects gaps between segments in a complex chain of elements. This error is common, but usually not serious.

The end points of elements comprising a complex chain do not always intersect, even when the component elements *appear* connected on the screen. Gaps may be revealed if the complex chain is dropped to its components.

When a TriForma file is detected, *FileFixer* will not report *or repair* error 555 unless you explicitly disable this setting. This ensures *FileFixer's* report emphasizes the most severe errors.

Do not check to see if the unused bits are set in the "properties" word.

FileFixer normally reports error 132 when it detects non-zero values in reserved bits (bits 4:6 and 8) in an element's properties. This error is not serious now, but because design file elements often have a very long life, it is possible that non-zero reserved bits will cause problems with future versions of MicroStation.

When a TriForma file is detected, *FileFixer* will not report error 132 unless you explicitly disable this setting. This ensures *FileFixer's* report emphasizes the most severe errors.

Additional enhancements in *FileFixer for V8* version 8.1a

When error 721 is detected, *FileFixer* now automatically handles by dropping the entire complex element into its component elements.

Elements manifesting error 365 can now be deleted automatically.

Improved robustness of *FileFixer's* analysis of structured storage errors.

FileFixer 8.1a was certified for MicroStation V8 versions 8.0.4.1 (third commercial release) and 8.1.0.7 (fourth commercial release).

8.0e — 23 May 2003

FileFixer 8.0e includes a number of V8-specific fine-tuning adjustments based on lessons learned since the first commercial release of *FileFixer for V8*.

FileFixer 8.0e enhancements include:

Elements manifesting error 185 (zero size in all dimensions) can now be deleted automatically from the non-model section of a file. This feature can be used, for example, to automatically delete empty shared cell definitions.

The first commercial version of *FileFixer for V8* (8.0c) was certified for the third commercial release of MicroStation V8 (8.0.4.1, July 2002). Clients have reported *FileFixer* works successfully with earlier versions of MicroStation V8, but for best results we recommend using *FileFixer for V8* with MicroStation 8.0.4.1 or later. *FileFixer* now issues a reminder when you use it with earlier versions of MicroStation V8.

FileFixer now properly processes nested reference files.

"Skip checking the file in Search mode when there is a seal-of-approval" now works consistently.

Implemented the search setting for error 119 (Element is too close to edge of design plane).

204 errors (missing graphical data in a model) are now suppressed for files with a ".cel" extension (because cell libraries commonly have an empty Default model).

304 (illegal H-bit), 555 (Line string is missing from complex chain) and 717 (class map is incorrect) errors are now consistently repaired.

FileFixer will now drop status on complex elements to handle 721 errors (illegal components in complex element).

FileFixer now correctly handles the component Count from Mesh Header elements.

FileFixer is now able to robustly report missing or damaged model/cell information [~Mh stream].

Miscellaneous user interface refinements.

MicroStation 8.0.4.1 incorrectly reports a class for non-graphical Smart Geometry elements (which do not include a display header). This is fixed in MicroStation 8.1.0.7. *FileFixer* reports the Smart Geometry class information provided by MicroStation.

Note: When necessary, *FileFixer* bypasses MicroStation and uses Axiom's file access techniques to analyze and repair V8 files. However, no attempt has been made to re-write MicroStation — *FileFixer* makes extensive use of MDL. Accordingly, as MicroStation V8 evolves, *FileFixer* benefits from Bentley MDL enhancements.

As a result, you may observe slight differences in *FileFixer's* analysis depending on your version of MicroStation V8. Our testing confirms we have handled the *major* differences between 8.0.4.1 (third commercial release) and 8.1.0.7 (fourth commercial release), but some minor differences might remain.

FileFixer was not certified for versions of MicroStation V8 before the third commercial release (8.0.4.1) — we strongly recommend upgrading to a later version of MicroStation V8.

FileFixer 8.0e was certified for MicroStation V8 versions 8.0.4.1 (third commercial release) and 8.1.0.7 (fourth commercial release).

8.0d — 17 December 2002

FileFixer 8.0d includes enhancements corresponding to *FileFixer* 7.8i (for V7). These enhancements include:

Optimized analysis of tag elements, reducing the amount of memory consumed.

Improved detection and handling of full or write-protected output disks.

FileFixer 8.0d also includes enhancements corresponding to *FileFixer* 7.8h (for V7). These enhancements include:

New Settings File Format: Plain Text

For many years *FileFixer* settings were saved in standard MicroStation resource files. Unfortunately, we were not able to make those resource files upwardly compatible with future versions of *FileFixer for V8*. As a result, customized *FileFixer* settings (changes from Factory Defaults) would have to be manually updated and saved each time a new version of *FileFixer* was installed. This was not ideal — we wanted to do something about it.

We are pleased to announce that *FileFixer*'s settings are now stored in a plain text file named `fixer.ini`. *FileFixer for V8* settings files will hereafter be upwardly compatible.

The first commercial release of *FileFixer for V8* (8.0c) saved its settings in MicroStation resource files. If you use customized settings (changes from the Factory Defaults) with *FileFixer* 8.0c, you will need to manually update your settings one more time when you install 8.0d. Your 8.0d settings will thereafter be upwardly compatible with future versions of *FileFixer for V8*.

Note: *FileFixer for V8* settings are quite different from *FileFixer* for V7 settings. Settings files are not interchangeable between the V7 and V8 versions of *FileFixer*.

Report Enhancements

Report file settings are now grouped in a single dialog selectable from *Settings* | *Change settings* | *Report File...*.

FileFixer now includes the name of the .fix or backup file in the output report. This documents the exact location and name of your files before and after repair.

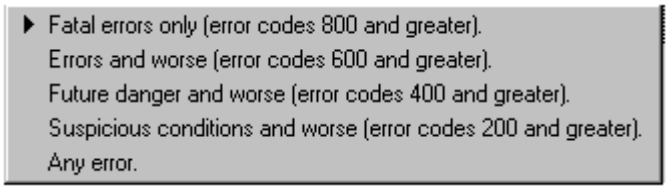
FileFixer now includes the MicroStation version at the end of each report to automatically document the version of MicroStation you are using with *FileFixer*. This is handy when clients send us files for analysis.

When processing a single file, *FileFixer*'s report file name now defaults to the directory and base name of the selected design file. When processing multiple files, the report file name now defaults to the *active* file's directory and the name "fixer.rep".

We enhanced and clarified reports. This is particularly noticeable in *FileFixer's* HTML reports.

Severity of Errors to be Reported

For clarity and consistency between the V7 and V8 versions of *FileFixer*, "Severity of Errors to be Reported" is now selectable by description and number, not just description:


Fatal errors only (error codes 800 and greater).
Errors and worse (error codes 600 and greater).
Future danger and worse (error codes 400 and greater).
Suspicious conditions and worse (error codes 200 and greater).
Any error.

TriForma Enhancements

FileFixer now accepts type 27 b-splines as components of type 18 surfaces and type 19 solids. These combinations of elements are normal constructs in TriForma files.

A new TriForma setting was added to accommodate unnamed cells nested within other unnamed cells. This setting is selectable from *Settings* | *Change Settings* | *MicroStation Applications* | *TriForma.* This setting is ON by default. *FileFixer* automatically applies this setting to TriForma files when TriForma element linkages are detected.

Miscellaneous Enhancements

Miscellaneous GUI refinements.

FileFixer now correctly calculates the element size for a file fence element (type 5, level 0, class 0).

FileFixer now correctly checks for error 383 when cells have a Words-in-description greater than 900.

FileFixer now only reports error 905 for b-spline poles when they are components of a complex header. Error 719 will now be reported for b-spline poles which are not components of a complex header. These error number re-assignments initiate the most optimum repair for each scenario.

Improved *FileFixer*'s handling of raster reference file attachments (type 90) when the Words-in-description (WID) was off by 1. *FileFixer* now repairs the WID value.

FileFixer 8.0d was certified for MicroStation V8 version 8.0.4.1 (third commercial release).

8.0c — 7 November 2002

The first commercial release of *FileFixer* for MicroStation V8.

FileFixer 8.0c includes the most popular V7 features adapted for V8, plus some V8-specific features.

FileFixer 8.0c was certified for MicroStation V8 version 8.0.4.1 (third commercial release).